Engineering with Computers
https://doi.org/10.1007/s00366-020-01179-5

ORIGINAL ARTICLE

Stochastic paint optimizer: theory and application in civil engineering

Ali Kaveh[1] · **Siamak Talatahari**[2] · **Nima Khodadadi**[1]

Received: 5 July 2020 / Accepted: 15 September 2020

Abstract
This paper presents an art-inspired optimization algorithm, which is called Stochastic Paint Optimizer (SPO). The SPO is a population-based optimizer inspired by the art of painting and the beauty of colors plays the main role in this algorithm. The SPO, as an optimization algorithm, simulates the search space as a painting canvas and applies a different color combination for finding the best color. Four simple color combination rules without the need for any internal parameter provide a good exploration and exploitation for the SPO. The performance of the algorithm is evaluated by twenty-three mathematical well-known benchmark functions, and the results are verified by a comparative study with recent well-studied algorithms. In addition, a set of IEEE Congress of Evolutionary Computation benchmark test functions (CEC-C06 2019) are utilized. On the other hand, the Wilcoxon test, as a non-parametric statistical test, is used to determine the significance of the results. Finally, to prove the practicability of the SPO, this algorithm is applied to four different structural design problems, known as challenging problems in civil engineering. The results of all these problems indicate that the SPO algorithm is able to provide very competitive results compared to the other algorithms.

Keywords Stochastic paint optimizer · Metaheuristic algorithm · Optimization

1 Introduction

For optimization problems, two major methods, containing mathematical and metaheuristic algorithms, are developed and applied; However, using mathematical algorithms is difficult and time-consuming for solving many optimization problems. Furthermore, they require a good starting point to successful converge to the optimum result, otherwise, they may be trapped in a local optimum. On the other hand, metaheuristic methods are often nature-inspired techniques that are able to explore the entire search space and exploit a final good result. Within an affordable computational time, they can find optimal or near-optimal solutions to the tough and even NP-hard problems. Unlike mathematical methods, they are very flexible and simple, making them popular among both researchers and practitioners [1]. Meta-heuristics, which are among the most promising and successful techniques [2], represent a family of approximate optimization techniques that have gained a lot of popularity in the past two decades. As aforementioned, these techniques rely on the rules observed in nature. The origins of inspiration can be divided into three classes:

i. Evolutionary Algorithms (EAs): The theories are based on biological evolution, such as interconnection, transition and selection (EAs) [3, 4]. Genetic algorithm (GA) [5, 6] is a popular EA describing chromosomal solutions composed of various genes. It seeks to obtain the strongest chromosome by repetitive processes of replication, mutation and selection.
ii. Swarm Intelligence Algorithms: These methods are a branch of computational intelligence that discusses the collective behavior emerging within self-organizing societies of agents. In the other words, Swarm intelligence was inspired by the collective behavior in societies in nature such as the movement of animals [7]. Animal behavior experiments have contributed to Ant Colony Optimization (ACO) [8] algorithm developed based on pheromone and biological ants in nature's food coordination strategies; Particle Swarm Optimization (PSO) [9] focused on bird or fish school-

✉ Ali Kaveh
alikaveh@iust.ac.ir

[1] School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran

[2] Department of Civil Engineering, University of Tabriz, Tabriz, Iran

Published online: 13 October 2020

ing's social behaviors. As another example, Artificial Bee Colony (ABC) [10] methods mimics the foraging behavior of honey bee colonies. Grey Wolf Optimizer (GWO) [11] algorithm applies to the hierarchy of command and the hunting process of Grey Wolves in nature. The interaction of various homogeneous agents in the environment results in mutual intelligence. The intelligence is decentralized, autonomous, and dispersed in an environment. These mechanisms are widely used in nature to tackle problems such as successful food forages, avoidance of predators, or relocation of the colony.

iii. Physical-based Algorithms: These algorithms take inspiration from physics. For instance, Charged System Search (CSS) [12] is developed according to the Coulomb law in electro-statics and the Newtonian mechanical laws, Big Bang – Big Crunch (BB-BC) [13] algorithm is based on the theory of Big Bang and Big Crunch. Colliding bodies optimization (CBO) [14] is a population-based stochastic optimization algorithm based on the governing laws of a one-dimensional collision between two bodies. Gravitational Search Algorithm (GSA) [15] is established based on the law of gravity and Simulated Annealing (SA) [16] is inspired by annealing in metallurgy.

A variety of new approaches have been developed in recent years and extended to many issues of multidisciplinary optimization. Some of these approaches are mentioned below:

Slime Mould Algorithm (SMA) [17], Water Strider Algorithm (WSA) [18], Fitness-Distance Balance (FDB) [19], Hybrid Invasive Weed Optimization-Shuffled Frog-Leaping Algorithm (SFLA-IWO) [20], Algorithm of the Innovative Gunner (AIG) [21], Red Deer Algorithm (RDA) [22] and Dragonfly Algorithm (DA) [23]. In addition, Kaveh et al. [24, 25] have made a major contribution to the development and application of modern metaheuristics.

On the other hand, art is utilized as the source of a few algorithms such as the Harmony search (HS) [26] and Color Harmony Algorithm (CHA) [27]. CHA is based on the Munsell, a color space that describes colors based on three perceptual attributes including hue, value and chroma. It uses eight different harmonic templates proposed by Matsuda [28, 29] based on an experimental investigation and it utilizes the modified definition of population diversity proposed by Cheng and Shi [30]. The memory in this algorithm plays a significant role. This algorithm has five different parameters that need to be adjusted. All these factors make the CHA difficult to implement a different type of optimization problems. On the other hand, the beauty of paints is utilized to propose a new optimization method in this paper. The new algorithm is based on art as a source of inspiration similar to CHA; however, the new algorithm needs no parameter or memory. Contrary to CHR, this method utilizes four simple combination mechanisms. In addition, borrow of color theory, basic techniques of combining colors are mathematically modeled to design the new algorithm as an optimization algorithm. Other key reasons for the development of the new algorithms are the advancement in computer technology, simplicity, flexibility and the need for art optimizers (especially for solving engineering problems).

It is notable that comparing to the random search methods, they are those stochastic methods that rely solely on the random sampling of a sequence of points in the feasible region of the problem, but this algorithm try to follow an optimization process, therefore it classifies the solutions into three groups (worst, good, best) in a logical manner, then utilizing some heuristic combination mechanisms are utilized for improving the existing solution. In other word, the algorithm is implemented in a way that it can find better and better solutions since the utilized mechanisms are simple.

The rest of the text is arranged as follows: Sect. 2 explains the origins of inspiration for the algorithm; Sect. 3 presents the formulations and overview of the new algorithm, so-called Stochastic Paint Optimizer (SPO). Several numerical mathematical and structural engineering problems are solved in Sect. 4. The numerical results obtained by different algorithms are discussed in Sect. 5. Finally, Sect. 6 concludes the findings and recommendations for future works.

2 Definitions and concepts

2.1 Background

Only a few numbers of algorithms are based on art. The art as the human activity for presenting his beautiful feels follows an optimization process. This means that an artist tries to present the best of he/she can generate by playing a music or painting a picture. Here, paints as one of the significant field of arts are utilized to proposing a new optimization algorithm. The specific colors in a paint, i.e. blue, green, yellow and red, associated with water, air, earth, and fire elements respectively, first identified by Leonardo da Vinci were a long-standing problem of visual perception, neurophysiology and cross-cultural linguistics [32]. The science and the art of using color are known as color theory. It describes how humans interpret color and the visual effects of color mixing, appropriate or contrasting. The color theory also includes the colors of the communicated message and its reproduction processes. Color theory is based on a color

wheel divided into three categories: primary, secondary and tertiary colors.

2.2 Color wheel

The color wheel has been used for many centuries not only to communicate various messages and religious but also to demonstrate basic laws of color association or mixture. To track their historical evolution, it may be helpful to differentiate between several forms based on fundamentally different relationships, using examples of late 19th-and 20th-century. The first type of seven colors containing red, orange, yellow, green, blue, indigo and purple is now generally referred to as the "additive" and is generally referred to as Newton's popular color circle published in 1704 [33]. Since that time, scientists and artists have studied and developed many variations in this definition. There is an ongoing debate on the differences of opinion about the superiority of one format over another. Harris [34] created the first printed circle of colors in 1766. He concluded that the colors of red, yellow and blue were the most distinct from each other and should be put as far apart as possible, separated by 120°, in a circle. LeBlon inspired Harris, who discovered the primary essence of red, yellow, and blue when combining pigments for printing. This revolving structure was taken over by Goethe [35]. While Goethe believed that there were two main colors (blue and yellow) and that all the colors originated from them, he was also highly influenced by LeBlon's predominant red–yellow–blue concept. Artists and painters still use the color wheel to develop color harmonies, mixing and palettes, as shown in Fig. 1. Colors on the color wheel have also been classified into three groups. According to this, Colors are clustered into three groups used in this work as follows (Fig. 2):

i. *Primary Colors:* These are three colors (red, yellow, blue) in traditional color theory that is not mixed with or influenced by any mixture of other colors. They are as the best colors (Cp_{best}), because all other colors are derived from these three colors.
ii. *Secondary Colors:* These are the colors formed by mixing the primary colors. They can be as the second best or good colors (Cs_{good}).
iii. *Tertiary Colors:* These are the colors made from primary and secondary colors, and their place is the worst ones (Ct_{worst}). These colors have a two-word name, such as blue-green, red-violet, and yellow-orange.

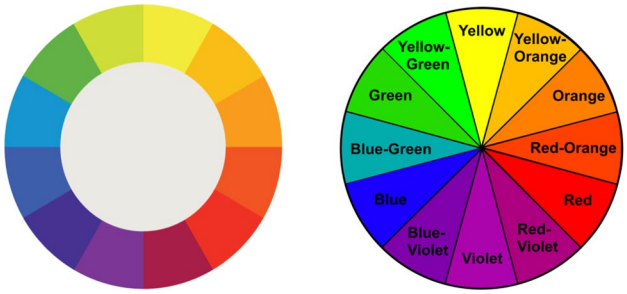

Fig. 1 Color wheel

Fig. 2 Color categories

2.3 Techniques for combining colors

The techniques for combining colors affects the spectator and provides an inner sense of order and a visual experience balance and it is boring and chaotic when everything is not a pleasing balance. This important point plays an important role in the proposed algorithm by balancing exploration–exploitation. Below are defined as the basic techniques for combining colors based on the color wheel. All these techniques are utilized to create a new colors (C_{new}).

2.3.1 Analogous combination technique

Analogous color schemes combine colors that are near to each other (neighborhood) on the color wheel, including purple-green, purple and yellow-orange next to each other. For forming an analog color, one color is selected as dominated, the other one as supports and the last one as accent. Typically, they suit well and plan calmly and conveniently as shown in Fig. 3. Mathematical point of view, the new colors can be defined using three other colors as C_{i-1}, C_i, and C_{i+1} in which i is the index of existing colors that can be selected in a random manner and in this way a stochastic selection can be guaranteed. In should be noted that one of these colors should be the dominated one and the rest are supports and accent. C_i is considered as the dominated in this paper and this means that the coefficient of this color is set to unity and its sign is positive while the coefficient of the other ones is a random vector. The sign of two other colors is positive and negative, respectively. This combination technique can be formulated mathematically as

$$C_{\text{new},1} = C_i + \text{rand}.\left(C_{i+1} - C_{i-1}\right) \tag{1}$$

It should be noted that all these three colors are belong to the same category of colors. Considering this point for using the above equation, we first should select a color category (Primary, Secondary or Tertiary) in a random manner and then in this category the dominated and related colors are selected.

Fig. 3 Analogous Combination Technique

For explaining the effect of this technique in the optimization process, one can use the graph theory concepts. the reason is that the graph theory provides a tool for explaining difficult concepts using just simple concepts. If the existing colors in the new algorithm are considered as nodes of a graph, the analogous combination technique create a new color (solution) using these three solutions that are near each other. The edges of the graph are those connect these three colors to each other. The neighborhood can be defined in different approaches. Here, the solution with close values of objective function is neighborhood. This means that this combination technique searches the neighborhood of the selected color (C_i) to exploit a little better color. In the other words, this technique treats as a local search approach and without considering that the select solution (C_i) is the best, intermediate or worst solution, it tries to improve its quality and in this way in initial iterations, it performs as a local searching while this is exchange to exploitation ability with progressing the searching process.

Although this technique is formulated by a simple equation (Eq. 1), it covers some important matters. If the selected category to be used by this technique is the primary, this technique directly attacks toward the so-far best solution for finding a near better solution. This point is important because without of attention to the number of the current iteration, this technique tries to improve the so-far best result and the success probability of this technique for the primary category is high; the reason is clear because it uses three good solutions for finding the better one. On the other hand, if the selected color is worst one (Tertiary), the technique strives to improve the worst result and, in this way, it helps the algorithm when using the other combination techniques. Finally, if Secondary category is selected, this technique tries to pick a solution to this category and add it to the primary one and in this way, the structure of color categories can be improved. To sum up, this one equation covers three different formulations for three predefined color categories.

2.3.2 Complementary combination technique

Colors opposite to each other on the wheel are known to be complementary colors (e.g. red and green) as presented in Fig. 4. The high contrast of complementary colors, especially when used with maximum saturation, produces a vivid look. It needs to be handled correctly so that the color scheme is not distracting. Mathematical point of view, this technique combines two opposite colors (say the best and the worst one) and add them to the existing color. Since there are three categories, the best one is Primary and the worst one is the Tertiary. As a result, this technique uses a random selected color from Primary category (C_{pi}), the other one a random selected color from Tertiary category (C_{Ti}) and finally the main one as an existing color (C_i). Similar to the

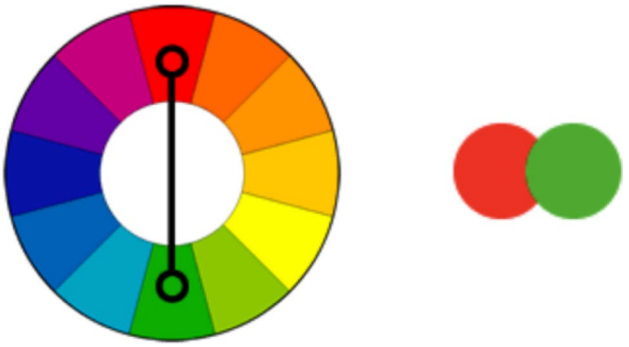

Fig. 4 Complementary Combination Technique

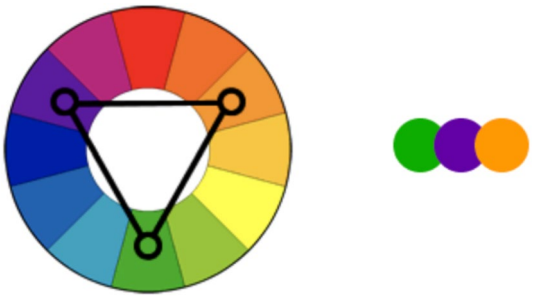

Fig. 5 Triad Combination Technique

analogous combination technique, the coefficient of the main color is set to one and a random vector is used for the other parts. As a result, the new color can be obtained as

$$C_{\text{new},2} = C_i + \text{rand}.\left(C_{\text{Pi}} - C_{\text{Ti}}\right) \tag{2}$$

The related graph for this technique is similar to the previous one with a difference that the C_{i+1} and C_{i-1} is replaced by C_{Pi} and C_{Ti}; however, optimization point of view this technique plays a completely different role. This technique tries to go toward one of the best solutions and go out of the worst ones, contemporaneity. Clearly, this technique covers the whole the search space and it treats as a strong global search technique. In the initial iterations of the optimization process that the best and worst solutions are far from each other, this technique tries to push the existing solutions toward to the best ones and goes away from the worst ones. In this way, the tendency of going toward the best solutions not only causes improving the existing solution but also explores the search space in an efficient approach. In the last iterations that the solutions become close to each other, this technique tries to approach the best and worst solutions to each other and in this way, it exploits some new better solution. as a result, this technique shows the exploration ability of the algorithm in the initial iterations while its exploitation ability becomes clear in the last iteration.

2.3.3 Triadic combination technique

A triadic color scheme is any three colors that are equally apart on the color wheel. For example, red, yellow and blue can be combined according to this technique as shown in Fig. 5. Triadic color schemes are very vivid, even though the colors are used light or unsaturated. Since three categories are defined in this paper, using this technique becomes simple. According to this technique, one should select three colors belonging to different categories (one from Primary, C_{pi}, the other from Secondary, C_{Si}, and the last one from Tertiary, C_{Ti}). Since these selected colors treats the same, using the mean of them can be useful. similar to previous techniques, a part of the mean of three colors and the whole of the main color are utilized to create the new color, as

$$C_{new,3} = C_i + \text{rand}.\left(\frac{C_{\text{Pi}} + \text{C}_{\text{Si}} + C_{\text{Ti}}}{3}\right) \tag{3}$$

Optimization point of view, this technique wants to use the mean of solutions for creating a new one. This mean is the mean of three random solutions (one form the best, one the intermediate and the last from the worst groups). This technique summarizes different states in one formula similar to the previous one. If the best solution (C_{pi}) is a strong solution, using just this term without the other ones absorbs the main solution (C_i) toward to this strong one and the searching process can not be performed in a complete form. It means that an optimization algorithm desires to search around the best solution rather than very close space of it. Therefore, the other terms help the algorithm to perform this important searching process. Here, the strong solution implies a very good solution with a considerable distance of other solutions. On the other hand, if the worst solution (C_{Ti}) is very weak one; similar to the previous discussion, a good algorithm tries to avoid it but since it is a very weak, it becomes difficult unless one uses others terms as presented in this algorithm. The weak solution implies a very bad solution with a considerable distance of other solutions. In addition, if these solutions are not very strong or very weak (they are still strong or weak), the above-mentioned point is still valid; however, its effect is reduced. In the other words, if the selected solutions are very strong (or weak) and their distance from other solutions are not small, this technique tries to present an exploration ability by proving a large searching process and if the selected solutions are not very strong (or very weak), the searching process are limited with a small part (exploitation). It should be noted that very strong (or very weak) results are generated in the initial iterations that the distance of solutions is high and the other ones are founded in the last iterations. As far as our best knowledge, a good algorithm should provide a good exploration at first

and increase the exploitation ability at the final iterations. This technique can handle this matter using all strong, weak and intermediate solutions.

2.3.4 Tetradic combination technique

Finally, the last combination technique used by painters is Tetradic. The four colors grouped in two complementary pairs of rectangular or tetradic colors are used, as depicted in Fig. 6. This rich color scheme provides a lot of variety. The color scheme works better if one color is dominant. This technique is mathematically modeled using the concepts of providing the previous techniques. The four color-categories are required for this technique; three of them can be selected form three pre-defined categories and the last one should be defined. The aim of this technique is to provide a strong stochastic searching that can help the algorithm when trapped in a local solution. Therefore, two differences are considered in this technique compared to the other ones:

1- Instead of using just one random vector, each color has its own random vector;

2- The fourth color is generated randomly.

These two points increase the exploration affect of this technique in comparing to the other techniques. Mathematically, it can be formulated as

$$C_{\text{new},4} = C_i + (\text{rand}_1.C_{\text{Pi}} + \text{rand}_2.C_{\text{Ti}} + \text{rand}_3.C_{\text{Si}} + \text{rand}_4.C_{\text{rand}})/4 \quad (4)$$

3 Stochastic paint optimizer

Using the concepts of colors in a paint, a new meta-heuristic algorithm known as Stochastic Paint Optimizer (SPO) is formulated and presented in this section.

3.1 Inspiration

Every paint needs different colors at its heart. The painters can generate these colors in endless ways. Quite often, an artist may be drawn toward a particular palette that tends to define the style of their entire body of work. Paint combination technique has been one of the keys to color practice, for painters in particular. Many people assume that the range of colors for a paint depends entirely on the taste and elegance of the painter. The method of color selection is, therefore, more complex than it appears and plays a pivotal role in the paints. One should understand how colors are created and how they relate to each other to produce a beautiful paint. That is why art schools, colleges and universities research color theory, which focuses on the essence of colors. In this paper definition of color theory, color wheel and color combination techniques are mathematically modeled to reach the Stochastic Paint Optimizer (SPO) as an optimization algorithm.

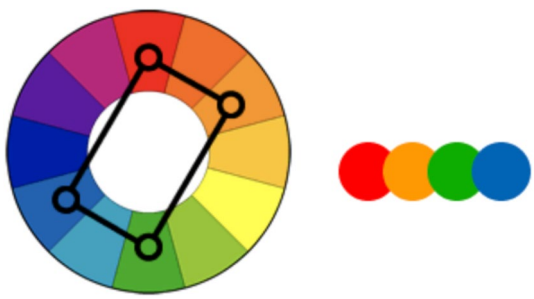

Fig. 6 Tetradic Combination Technique

3.2 Mathematical model and algorithm

The main steps of this algorithm contain creating initial paints, paint clustering, paint combination, and stop controlling. The search space is specified as a canvas and paints as solutions that contain some colors as design variables. Paints are evaluated and sorted in increasing order according to their corresponding their beauty index (objective function values). Any new color that is applied to canvas is an integral part of the piece's perception. Due to this, each color has its grades (values) according to color wheel categories primary (the best), secondary (good), and tertiary (the worst) colors. Thanks to these equal categories, there is no need to add parameters in the algorithm. Using provided combination techniques to make new colors, this algorithm can create optimum paints (or solutions).

3.3 Main steps of the stochastic paint optimizer

Figure 7 displays the flowchart of the SPO and the main steps of the algorithm are as follows:

Phase 1: Initialization

In an *nc*-dimensional search object, the initial colors of all paints are determined, randomly.

$$C_{i,0} = C_{\min} + \text{rand}.\left(C_{\max} - C_{\min}\right),\ i = 1, 2, 3, ..., nc \quad (5)$$

where, $C_{i,0}$ is the initial color of the i th paint. $C_{\min}$ and $C_{\max}$ are the lower and upper limits of the design variable i, *rand* is a random number with its range [0, 1] and *nc* is the number of colors or variables. It should be noted that all colors with each other generate a paint that is a design or solution of optimization problems. Then, Evaluation of the objective function for each paint is gained. In this way the beauty of each paint becomes clear.

Phase 2: Evaluation, Sorting and Clustering

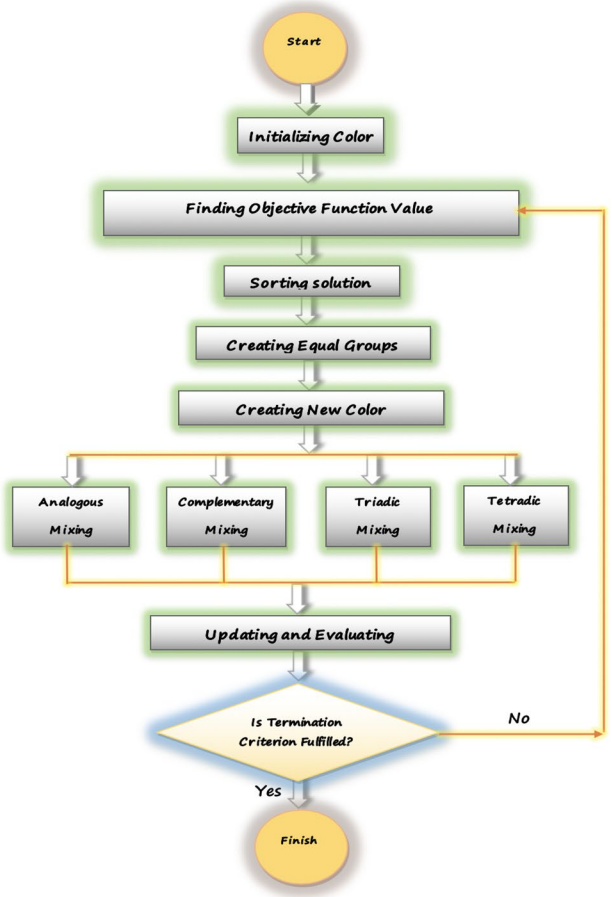

Fig. 7 Flowchart of the SPO algorithm

Paints as the result of the problem are sorted in increasing order according to their corresponding objective function. Finally, they are clustered into three equal groups namely, primary (the best), secondary (good), and tertiary (the worst) as mentioned in Sect. 2. In this way, there is no need to add parameters in the algorithm for clustering.

Phase 3: Utilizing Combination Techniques

In this step, new paints are generated using four different combination techniques provided in Sect. 2.3.

Phase 4: Evaluating and Updating.

The new paints are evaluated and if their new beauty index is better than the previous one, it is replaced instead of the old one.

Phase 5: Checking Termination

Following a set of iterations, the optimization cycle is terminated. If the criterion is not met, a new process is scheduled for phase 2; otherwise, the process ceases and the best solution will be reported.

Algorithm 1: Pseudo-code of SPO

Inputs: The Number of Paints (Paint size)

Outputs: The best Paint and its objective value

Initialize the random paints as Eq. (5)

Evaluate and sort the paints according the pre-defined objective function

While The final criteria are not met

  Cluster the paints to three categories as Primary, Secondary and Tertiary

    for every paint

    create new paint (combination techniques) Eqs. (1)-(4)

    End for

  Evaluate the fitness of all paints

  Update each paint if the new is better than the previous one

End while

Return the best result

3.4 Similarities and differences of the SPO and other algorithms

Recently, many new algorithms are presented as metaheuristics; similarities of them are:

- They try to follow an optimization process using a clever and random selection. In this regard, they use random numbers and generate a new solution in each step.
- Many of the metaheuristics are multi-agents. This means they generate several solutions at the same time rather than one.
- Comparing the generated solution with each other is the main key to clear the next step.
- Almost, all algorithms need to determine the maximum number of iterations in each iteration. Some algorithms need more other parameters that should be defined carefully, however, the present method does not need additional parameters.

The differences in this method with the new SPO are:

- In the new algorithm, solutions (paints) are clustered into three groups namely, primary (the best), secondary (good), and tertiary (the worst); while most of the other methods do not divide the solutions.
- New solutions are generated over the entire search space using four different techniques. Many methods use just one or two strategies for generating the new solutions.
- Creating Vicinity colors step is responsible for generating similar solutions to improve the previously obtained solution, and this approach contributes to good convergence.
- This method not only use a separate plan for using best (primary), good (secondary) and the worst (tertiary) solu-

Table 1 The unimodal benchmark functions

Function	Dim	Range	$f_{\min}$
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[− 100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[− 10,10]	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_i^2\right)^2$	30	[− 100,100]	0
$F_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30	[− 100,100]	0
$F_5(x) = \sum_{i=1}^{n} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[− 30,30]	0
$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[− 100,100]	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0,1)$	30	[− 1.28,1.28]	0

tion separately but also combining different categories of the solution is also considered. In this way controlling the exploitation and exploration of the algorithm becomes feasible. The main difficulty of many other meta-heuristics is at this point.

– Providing a simple process, however useful one, is the main feature of this algorithm; many other methods utilize some difficult idea and technique that become them hard to use in different fields.

4 SPO validation

The validation of the new algorithm is presented in this section consisting of two parts; Sect. 4.1 includes a broad variety of complex mathematical problems to be evaluated, with results in contrast with other metaheuristic methods and Sects. 4.2 utilizes some well-known structural engineering problems.

4.1 Mathematical benchmark problems

In this section, the performance of the new algorithm is evaluated using some unimodal and multimodal functions. Then, the convergence rate, positions of colors, and one non-parametric test for SPO are examined. The experiments are evaluated in two types as classical and CEC test examples.

4.1.1 Classical benchmark test functions

A list of 23 popular, unimodal, multimodal, and fixed-dimension multimodal benchmarks has been examined [36–39]. Tables 1, 2, 3 lists these functions. In these tables, *dim* is the dimension of functions; boundary denotes the upper and lower limits of search space and $F_{\min}$ stands for the optimal values of the functions. The number of separated runs is set to 30 for each function. Every value below $1E-30$ is recorded as 0. For a fair comparison, for all algorithms, 5000 multipled by dimension is set as the maximum number of function evaluations. SPO has also been checked under the same conditions to ensure comparative accuracy with other algorithms. Table 4 lists control and specific parameter settings for the utilized algorithms.

Tables 5, 6, 7 show the statistical results for the unimodal, multimodal and multimodal with the fixed dimension, respectively. The results of the SPO algorithm is compared to DA [23] as a swarm-based and GSA [40] as a physics-based algorithm. In addition, it is compared with some newly developed algorithms containing Grey Wolf Optimizer (GWO) [11], Sine Cosine Algorithm (SCA) [41], Water Strider Algorithm (WSA) [18], Moth-Flame Optimization

Table 2 The multimodal benchmark functions

Function	Dim	Range	$f_{\min}$
$F_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	[− 500,500]	− 418.9829×5
$F_9(x) = -\sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[− 5.12,5.12]	0
$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	[− 32,32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[− 600,600]	0
$F_{12}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\}$ $+\sum_{i=1}^{n} u(x_i, 5, 100, 4), y_i = 1 + \frac{x_i+1}{4}$	30	[− 50,50]	0
$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
$F_{13}(x) = 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[− 50,50]	0

Table 3 Multimodal benchmark functions with fixed-dimension

Function	Dim	Range	f_{min}
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[65.536, − 65.536]	1
$F_{15}(x) = \sum_{i}^{11}\left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]^2$	4	[− 5,5]	0.00030
$F_{16}(x) = 4{x_1}^2 - 2.1{x_1}^4 + \frac{1}{3}{x_1}^6 + x_1x_2 - 4{x_2}^2 + 4{x_2}^4$	2	[− 5,5]	− 1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}{x_1}^2 + \frac{5}{\pi}x_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	[− 5,5]	0.398
$F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3{x_1}^2 - 14x_2 + 6x_1x_2 + 3{x_2}^2)\right] \times\left[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12{x_1}^2 + 48x_2 - 36x_1x_2 + 27{x_2}^2)\right]$	2	[− 2,2]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$ $a = \begin{bmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}$ and $c = \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix}$ $p = \begin{bmatrix} 0.3689 & 0.117 & 0.2673 \\ 0.4699 & 0.4387 & 0.747 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{bmatrix}$	3	[0,1]	− 3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$ $a = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 17 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}$ $c = \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix}$ and $p = \begin{bmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{bmatrix}$	6	[0,1]	− 3.32
$F_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	− 10.1532
$F_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	− 10.4028
$F_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	− 10.5363

Table 4 Parameter settings of the utilized algorithms

Algorithm	Parameters
GSA	Population number (Np) = 50 Gravitional initial value (G_0) = 100 coefficient (a) = 20
GWO	Population number (Np) = 50 coefficient vector C = [0,1] A = [−1, 1] a = [0,2]
SCA	Population number (Np) = 50 coefficient (a) = 2
MFO	Population number (Np) = 50 logarithmic spiral shaped (b) = 1 Random number (t) = [−1.1] Convergence constant (r) = [−1,2]
DA	Population number (Np) = 50. Interia weight (w) = [0.9, 0.2] Separation weight (s) = 0.1. Alignment weight (a) = 0.1, Cohesion weight (c) = 0.7 Food factor (f) = 1 Enemy factor (e) = 1
MVO	Population number (Np) = 50 wormhole existence probability (WEP) : WEP_{max} = 1, WEP_{min} = 0.2 coefficient (P) = 6
WSA	Population number (Np) = 50 territory size (Pt) = 25 Attraction probability (Pro) = 0.5
SPO	Population number (Np) = 20

(MFO) [42] and Multi-Verse Optimizer (MVO) [43]. The results reveal that the total score of SPO was 26, which is the best among the other utilized algorithms as shown in Table 8. The mean ranks of algorithms are summarized in Fig. 8 for statistical measures like mean and standard deviation (SD). Clearly in terms of statistical ranks, the SPO is the best one compared to the other considered algorithms.

Figure 9 present the rank of the algorithms for the unimodal functions (F1–F7 functions) and the multimodal ones (F8-23 functions). The results of the unimodal functions indicate that SPO is capable of exploiting the space efficiently as shown in Fig. 9a,b. SPO is ranked the first in four of seven unimodal functions. The algorithm outperforms all others as well. It is worth noting that unimodal functions

Table 5 The statistical results of benchmark functions F1-F7

No		GSA	GWO	SCA	MFO	DA	MVO	WSA	SPO
F1	AVE	2.53E−16	6.59E−28	1.94E−16	2000	101.3336	0.022462	0*	0*
	SD	9.67E−17	6.34E−05	9.45E−16	4068.381	96.58759	0.006160	**0**	**0**
F2	AVE	0.055655	7.18E−17	1.28E−18	28.66667	7.530338	8.877180	4.89E−28	0*
	SD	0.194074	0.029014	4.59E−18	15.69831	6.143436	33.74294	1.94E-27	0
F3	AVE	896.534	3.29E−06	650.4789	16,833.44	6410.172	1.916605	0.014089	6.105E-11
	SD	318.9559	79.14958	1248.101	12,520.75	5456.428	0.706664	0.011180	8.055E-07
F4	AVE	7.35487	5.61E−07	1.335270	45.80769	5.930244	0.218609	0.000490	0.0020
	SD	1.741452	1.315088	1.828663	13.34950	7.065519	0.093077	0.000354	0.0028
F5	AVE	67.54309	26.81258	27.51712	18,268.11	2890.214	176.5168	32.42146	6.9216
	SD	62.22534	69.90499	0.556120	36,497.23	3939.582	253.5243	29.52849	17.9523
F6	AVE	2.5E−16	0.816579	3.720126	1340.033	127.2942	0.018740	0*	0*
	SD	1.74E−16	0.000126	0.334225	3474.982	101.2584	0.005324	**0**	**0**
F7	AVE	0.089441	0.002213	0.00598	1.375182	0.067759	0.005043	0.006433	0.00526
	SD	0.04339	0.100286	0.00590	3.577760	0.050227	0.001798	0.001839	0.00416

Bold values are related to the best results obtained by the selected algorithms

* Algorithm found Global Optimum

Table 6 The statistical results of benchmark functions F8-13

No		GSA	GWO	SCA	MFO	DA	MVO	WSA	SPO
F8	AVE	− 2821.07	− 6123.1	− 4314.06	− 8732.63	− 6793.96	− 7918.0	− 9354.74	− 2634.73
	SD	493.0375	4087.44	255.6938	1072.626	989.3598	782.706	653.1757	**375.3053**
F9	AVE	25.96841	0.310521	1.83186	148.5348	60.4525	112.474	40.56002	35.7204
	SD	7.470068	47.35612	6.96691	40.87507	27.58083	35.2363	10.78416	10.8194
F10	AVE	0.062087	1.06E-13	12.38263	9.7306	5.09027	0.15472	1.88E-14	0.2067
	SD	0.23628	0.077835	9.10541	9.74539	2.14984	0.40645	**4.52E-15**	0.4319
F11	AVE	27.70154	0.004485	0.00786	21.14255	1.9369	0.10271	0.016042	0.0042
	SD	5.040343	0.006659	0.02831	45.56648	1.57439	0.03706	0.020111	0.0064
F12	AVE	1.799617	0.053438	0.3975	0.25016	2.68162	0.20096	0*	0*
	SD	0.95114	0.020734	0.13269	0.48244	5.13881	0.37664	**0**	**0**
F13	AVE	8.899084	0.654464	2.06867	1.37E+7	10.19126	0.01012	0*	0*
	SD	7.126241	0.004474	0.13673	7.49E+7	11.45285	0.01326	**0**	**0**

Bold values are related to the best results obtained by the selected algorithms

* Algorithm found Global Optimum

are suitable for evaluating exploitation capacity of the algorithm. As a result, these findings demonstrate SPO's superior efficiency in terms of exploiting the optimum. Exploration, on the other hand, is the search part that focuses on the unexploited search areas. The SPO explores the search space efficiency as well. To have a detailed view, investigation of the results of the multimodal functions with a lot of local optima that its number is exponentially increased with dimension, are suitable for checking the exploration ability of an algorithm. According to this, statistical results of the multimodal functions (F8–F23) indicate the high exploration capability of SPO, where the first rank is gained by this method as depicted in Fig. 9c,d. This method outperforms GWO, SCA, MFO, DA and MVO on the majority of these functions. Moreover, SPO indicates very competitive results compared to WSA and GSA; and outperforms them as well.

4.1.2 CEC-C06 2019 benchmark test functions

As an additional evaluation of the SPO, a group of 10 modern CEC-C06 2019 benchmark test functions are used. These functions are enhanced for a single objective optimization problem [44] and are referred to as 'The Task of 100 Digits' which are intended for annual optimization competitions. Functions CEC04 to CEC10 are shifted and rotated, while functions CEC01 to CEC03 are not. Table 9 shows the details of these functions. In addition, all test functions are scaled. Five modern and well-cited algorithms, DA [45],

Table 7 The statistical results of benchmark functions F14-23

No		GSA	GWO	SCA	MFO	DA	MVO	WSA	SPO
F14	AVE	5.859838	4.042493	1.794415	1.525135	1.757204	1.560495	0.998004	0.9985
	SD	3.831299	4.252799	1.892839	1.34095	1.289434	0.810885	**1.13E−16**	5.521E-04
F15	AVE	0.003673	0.000337	0.001134	0.00092	0.001832	0.003426	0.000549	0.0003074*
	SD	0.001647	0.000625	0.00035	0.000284	0.001337	0.006762	0.00032	**1.581E−08**
F16	AVE	−1.03163*	−1.03163*	− 1.03156	−1.03163*	−1.03163*	−1.03163*	−1.03163*	−1.03163*
	SD	**4.88E−16**	1.03163	7.79E−05	6.78E−16	2.70E−14	1.47E-06	5.68E−16	7.87E−06
F17	AVE	0.397887*	0.397889	0.403152	0.397887*	0.397887*	0.39789	0.397887*	0.397887*
	SD	**0**	0.397887	0.007668	**0**	6.48E-15	3.98E-06	**0**	**0**
F18	AVE	3*	3.000028	3.000109	3*	3*	3.000014	3*	3*
	SD	4.17E−15	3	0.000124	2.04E−15	4.13E−09	1.23E−05	2.91E−15	1.676E−14
F19	AVE	−3.86278*	−3.86263	− 3.85388	−3.86278*	−3.86071	−3.86278*	−3.86278*	−3.86278*
	SD	**2.29E−15**	3.86278	0.002132	2.71E−15	0.003279	2.83E−06	2.46E−15	2.25E−08
F20	AVE	− 3.31778	− 3.28654	− 2.94575	− 3.22824	− 3.23633	− 3.25038	− 3.25066	−3.3204*
	SD	0.023081	3.25056	0.320805	0.053929	0.081325	0.059472	0.059241	**3.164E−04**
F21	AVE	− 5.95512	− 10.1514	− 3.37565	− 7.30772	− 6.01288	− 7.80433	− 6.72819	− 9.4450
	SD	3.737079	9.14015	2.046341	3.400748	2.150163	3.018852	3.378711	1.6558
F22	AVE	− 9.68447	− 10.4015	− 4.06593	− 8.17415	− 6.47641	− 8.32628	− 7.35819	−10.4028*
	SD	2.014088	8.58441	1.942808	3.250245	2.735904	3.062084	3.609873	3.5E−05
F23	AVE	− 10.5346	− 10.5343	− 4.667	− 8.66814	− 6.24959	− 9.02186	− 8.30703	− 10.5190
	SD	**2.6E−15**	8.55899	1.758723	3.183513	2.475969	2.584132	3.499898	1.3832

Bold values are related to the best results obtained by the selected algorithms

* Algorithm found Global Optimum

Table 8 Algorithm performance comparison on benchmark functions 1–23

Algorithm	GSA	GWO	SCA	MFO	DA	MVO	WSA	SPO
No. found global optimum	4	1	0	4	3	2	8	**12**
No solution better than others	5	4	0	4	4	2	9	**14**
Total score	9	5	0	8	7	4	17	**26**

Bold values are related to the best results obtained by the selected algorithms

WOA [46], BOA [47], CSO [48] and SSA [49], are selected to compare their results to the SPO. Table 10 presents the outcomes of all algorithms in the form of mean and standard deviations. The results show that the SPO algorithm can produce highly satisfactory results and it is ranked first in four of ten CEC-C06 benchmark functions.

4.1.3 Convergence behavior of SPO

A significant feature of metaheuristics is convergence speed or convergence rate. According to Berg et al. [50], abrupt changes in search agent movement should occur during the initial optimization steps. This allows a metaheuristic to thoroughly explore the search space. Then, these changes should be reduced to emphasize exploitation at the end of optimization [11]. The convergence cure of all 23 benchmark functions is shown in Fig. 10. Obviously, in many problems, the new algorithm has the highest rate of convergence. Besides, in the first, second, and fourth column of Fig. 10, perspective views, contour views, search color history on all 23 benchmark functions are shown, respectively. The initial steps of iterations change suddenly, which are slowly decreased during this time. This conduct will ensure that the algorithms ultimately converge to a point in search space according to Berg et al. [50].

Position of new solutions in different iterations can provide an insight into convergence behavior of the algorithm. To fulfill this aim, the function nine (F9) as the test function with many local optimums is selected to be investigated with more details. The paints (solutions) are distributed randomly in the search space (canvas) at the beginning of the optimization. Figure 11 indicates the current positions of paints during the iteration phases. The solutions are collected almost after 20 iterations on the core sections that have almost optimum objectives. Some solutions display some local optimal in iteration 40, and some others explore the search space yet. Most of the

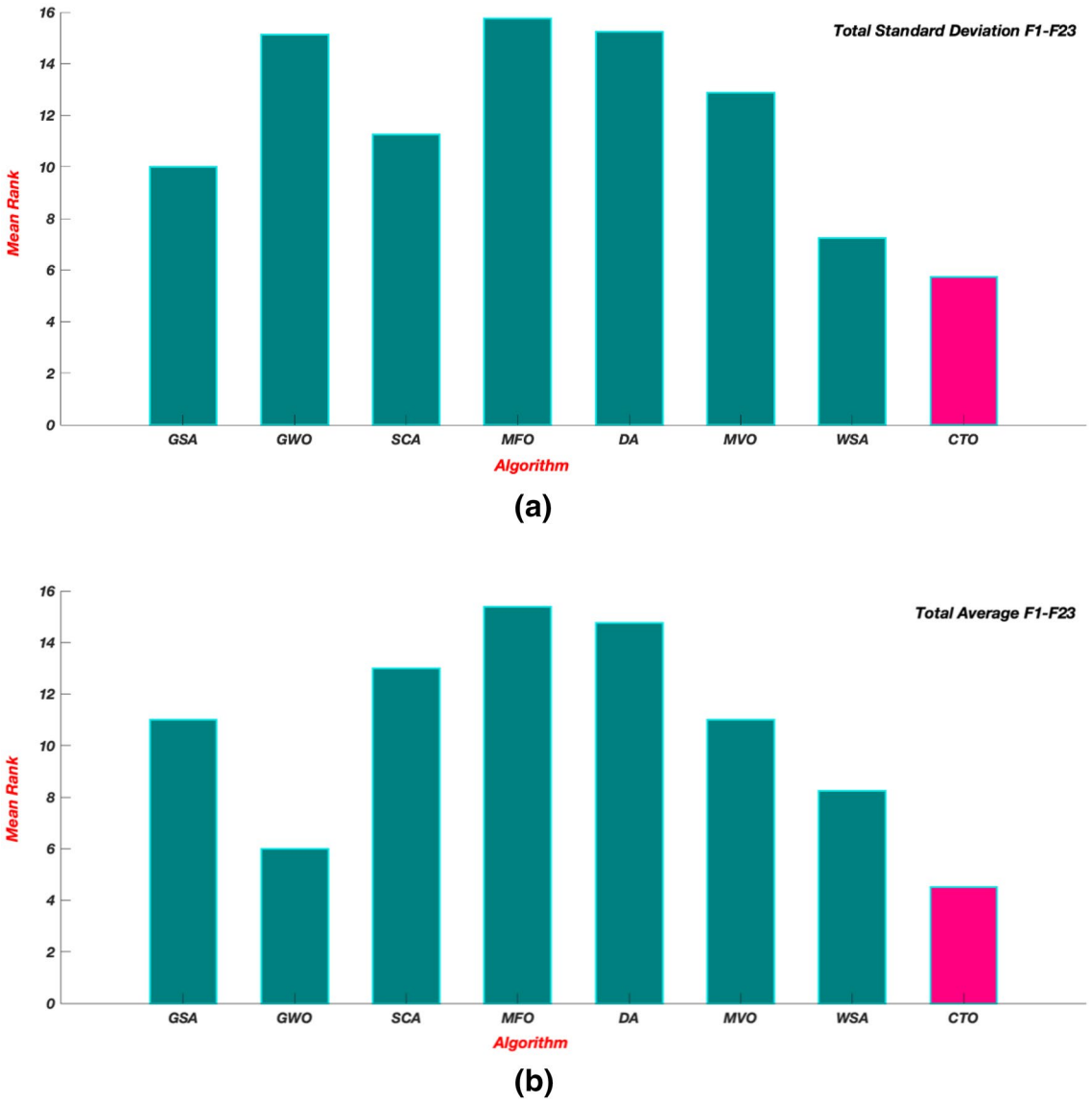

Fig. 8 Comparison of statistical ranks for 23 benchmark functions: (**a**) Total standard deviation and (**b**) Total Average

solutions are located on the local optima at iteration 60. Iteration 80 shows that they all are located near the global optimum result. Finally, the optimal point in iteration 100 is exploited by the paints.

4.1.4 Non-parametric Wilcoxon test

The comparison of algorithms based on mean and standard deviation (statistical criteria), does not usually compare all runs since the superiority is still possible. Therefore, a non-parametric regression test was conducted to compare the outcomes of each run and decide on the importance of the outcomes. In this work, the Wilcoxon test was used to determine the significance of the results as a non-parametric statistical test. The $p-values$ at 0.05 obtained from this test are illustrated in Table 11, which demonstrates the noticeable superiority of the SPO to other algorithms based on $p-values$, which are of less than 0.05.

4.2 Structural problems for SPO

Optimization of the structures involves achieving optimal cross-section (A_i) values that minimize structural weight (W). This minimum design also has to follow the numerous requirements that limit parameter design sizes and structural responses:

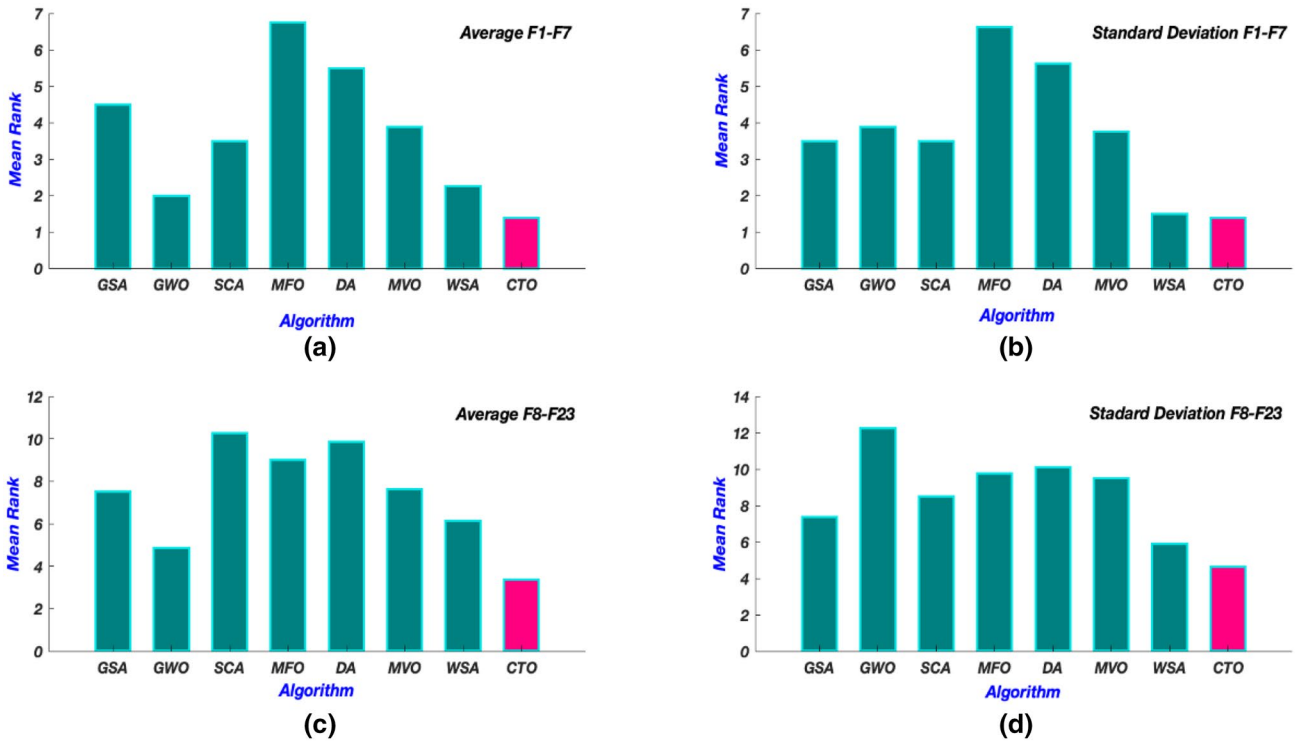

Fig. 9 Comparison of mean ranks: (**a**) average of F1-F7 functions, (**b**) standard deviation of F1-F7 functions, (**c**) average of F8-23 functions and (**d**) standard deviation of F8-F23 functions

Table 9 CEC-C06 2019 benchmark functions

No	Function	Dim	Range	f_{min}
CEC1	Storn's chebyshev polynomial fitting problem	9	[− 8192, 8192]	1
CEC2	Inverse hilbert matrix problem	16	[− 16384, 16384]	1
CEC3	Lennard–Jones minimum energy cluster	18	[− 4,4]	1
CEC4	Rastrigin's function	10	[− 100, 100]	1
CEC5	Griewangk's function	10	[− 100, 100]	1
CEC6	Weierstrass function	10	[− 100, 100]	1
CEC7	Modified Schwefel's function	10	[− 100, 100]	1
CEC8	Expanded Schaffer's f6 function	10	[− 100, 100]	1
CEC9	Happy cat function	10	[− 100, 100]	1
CEC10	Ackley function	10	[− 100, 100]	1

Bold values are related to the best results obtained by the selected algorithms

$$\text{Find } \{X\} = [x_1, x_2, \ldots, x_{ng}]$$

$$\text{Minimize } \mathrm{W}(\{\mathrm{x}\}) = \sum_{i=1}^{nm} \gamma_i, A_i, L_i(x)$$

$$\text{Subjected to} \begin{cases} g_j(\{X\}) \le 0 j = 1, 2, 3, \ldots, ns \\ A_{\min} \le A_i \le A_{\max} \end{cases} \tag{6}$$

where $\{X\}$ the vector of design variables, ng is the number of design variables, the structure weight is described as $W(\{x\})$, the number of structural members is presented by nm and the number of the constraints is described as ns. Here, the member's material density is presented by γ_i, the member's length is L_i, A_i is the member's cross-sectional area and $g_j(\{X\})$ denotes design constraints, respectively. The well-known penalty function due to the simple principle and ease of implementation is utilized for handling the constraints as:

$$f_{penalty}(X) = (1 + \varepsilon_1 . v)^{\varepsilon_2}$$
$$v = \sum_{i=1}^{n} \max[0, v_i] \tag{7}$$

Table 10 CEC-C06 2019 benchmark test results

No		CSO	DA	BOA	WOA	SSA	SPO
CEC1	AVE	1.58E+09	3.8E+10	**58,930.69**	411E+08	605E+07	6.3E+08
	SD	1.71E+09	4.03E+10	**11,445.72**	542E+08	475E+07	2.79E+08
CEC2	AVE	19.70367	83.73248	18.91597	17.3495	18.3434	**17.34285**
	SD	0.58672	100.1326	0.291311	0.0045	0.0005	**0**
CEC3	AVE	13.70241	13.70263	13.70321	13.7024	13.7025	**12.70240**
	SD	2.3E-06	0.000673	0.000617	**0**	0.0003	1.12E-09
CEC4	AVE	179.1984	371.2471	20,941.5	394.6754	**41.6936**	50,026.96
	SD	55.37332	420.2062	7707.688	248.5627	22.2191	**11.89894**
CEC5	AVE	2.671378	**2.571134**	6.176949	2.7342	2.2084	6.375451
	SD	0.171923	0.304055	0.708134	0.2917	0.1064	**0.011634**
CEC6	AVE	11.21251	10.34469	11.83069	**1.0325**	6.0798	9.133086
	SD	0.708359	1.335367	0.771166	1.0325	1.4873	**0.610547**
CEC7	AVE	365.2358	534.3862	1043.895	490.6843	410.3964	**251.9751**
	SD	164.997	240.0417	215.3575	194.8318	290.5562	**43.37608**
CEC8	AVE	**5.499615**	5.86374	6.337199	6.909	6.3723	5.842824
	SD	0.484645	0.51577	0.359203	0.4269	0.5862	**0.164953**
CEC9	AVE	6.325862	8.501541	2270.616	5.9371	**3.6704**	4294.314
	SD	1.295848	16.90603	811.4442	1.6566	0.2362	**2.77E-12**
CEC10	AVE	21.36829	21.29284	21.4936	21.2761	21.04	**20.25560**
	SD	0.06897	0.176811	0.079492	0.1111	0.078	**0.056284**

Bold values are related to the best results obtained by the selected algorithms

where, the sum of the constraints broken is v and constants ε_1 and ε_2 are selected considering the exploration and the exploitation rate of the search space. Here,ε_1 is set to 1, ε_2 is selected in a manner to decrease the penalties and reduce the cross-sectional areas. Then, ε_2 is set to 1.5 at the start of the search process and is eventually elevated to 3. This simple penalty function for handling the constraints is selected. The reasons are that it is simple principle and ease to implement. In addition, it has a small influence on the performance of the algorithm and in this way the evaluating the algorithm can be fair.

This section presents four structural optimization problems and their optimum designs obtained by the SPO algorithm and compares those obtained by some other methods from literature. The results are based on 30 independent runs and 20 initial population size the same as the previous section. SPO is parameter-free, and only a change in population size is required.

4.2.1 A 52-bar planar truss structure

The details of 52-bar planar truss structures with twenty nodes are shown in Fig. 12. The members of this truss are classified into twelve groups. The density of the material is 7860 kg/m^3 with the elasticity module of 2.07×10^5MPa. The maximum allowable stress in terms of both tension and compression for all members is 180MPa. Loads $P_x = 100$kN and $P_y = 200$kN are both applied in four top nodes. Discrete variables are chosen from Table 12. Table 13 displays the results of the SPO algorithm, as well as other forms of optimization. It can be seen the MBA [54], SOS [55] and SPO algorithms can find better results than the GA [51], HS [52], and IACO [53]. The statistical results in terms of the mean, best solution and standard deviation are 1903.65 kg, 1902.605 kg and 2.975 for the new algorithm, respectively which are the best among the other methods. Figure 13 depicts the best and average convergence history for the SPO algorithm and some other methods. As it can be seen the best curve belongs to the SPO, and even the average history of the new algorithm performs better than the best history of some other utilized algorithms. The existence stress of the best solution of the SPO algorithm is shown in Fig. 14 that proves the feasibility of the final result.

4.2.2 A 120-bar dome truss structure

Soh and Yang [56] first studied the 120-bar dome truss to achieve the optimum size and then many researchers utilized it to evaluate their methods. In this paper, we select this well-known example as the second structure. The density of the material and its elasticity module are 0.288 lb/in^3(7971.810kg/m^3)and30450 ksi(2.1×10^5MPa), respectively and the yield stress is taken as 58 ksi(400 MPa). Figure 15. displays all information about the

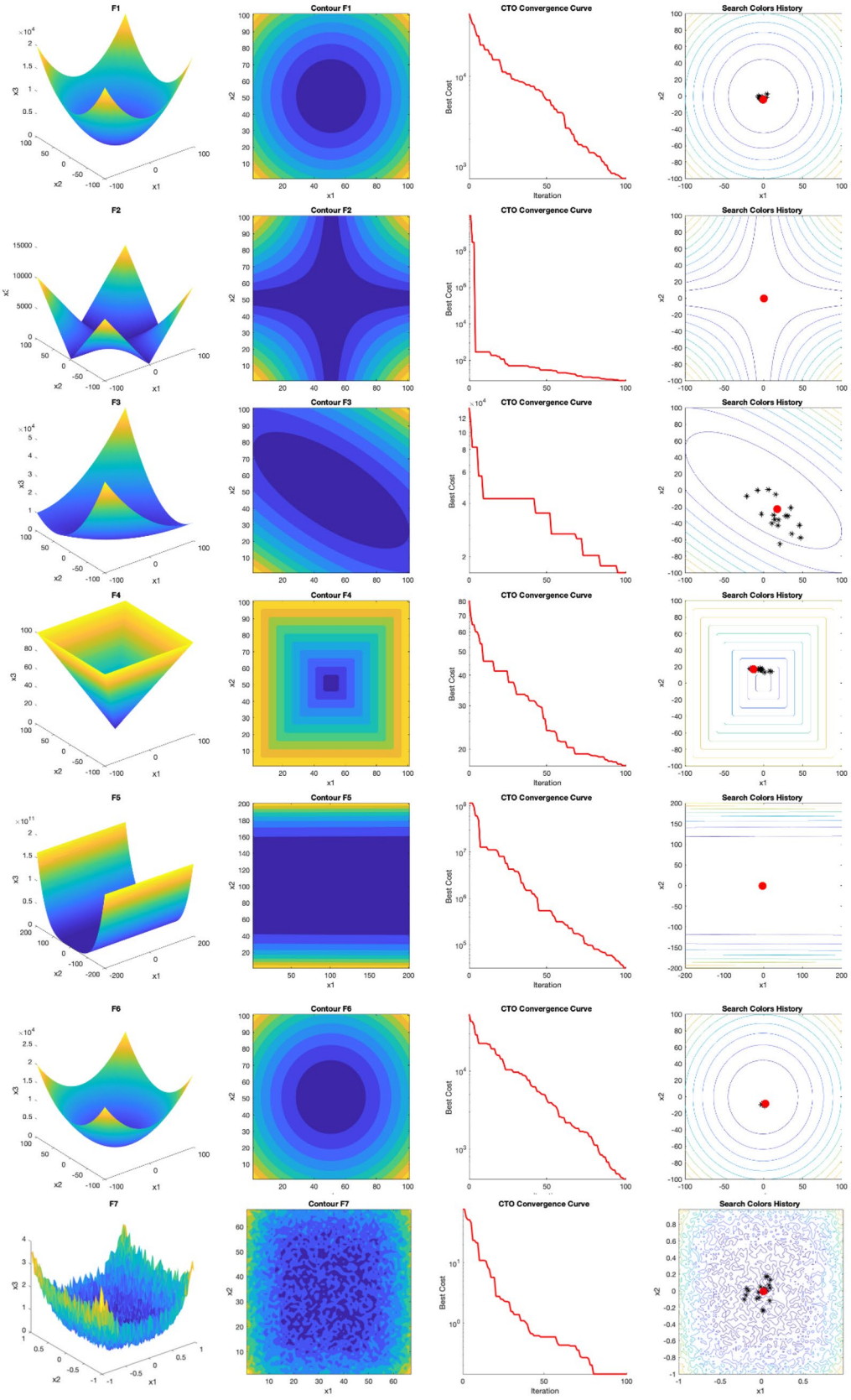

Fig. 10 Perspective views, counter view, convergence curves, convergence color positions of 23 benchmark functions

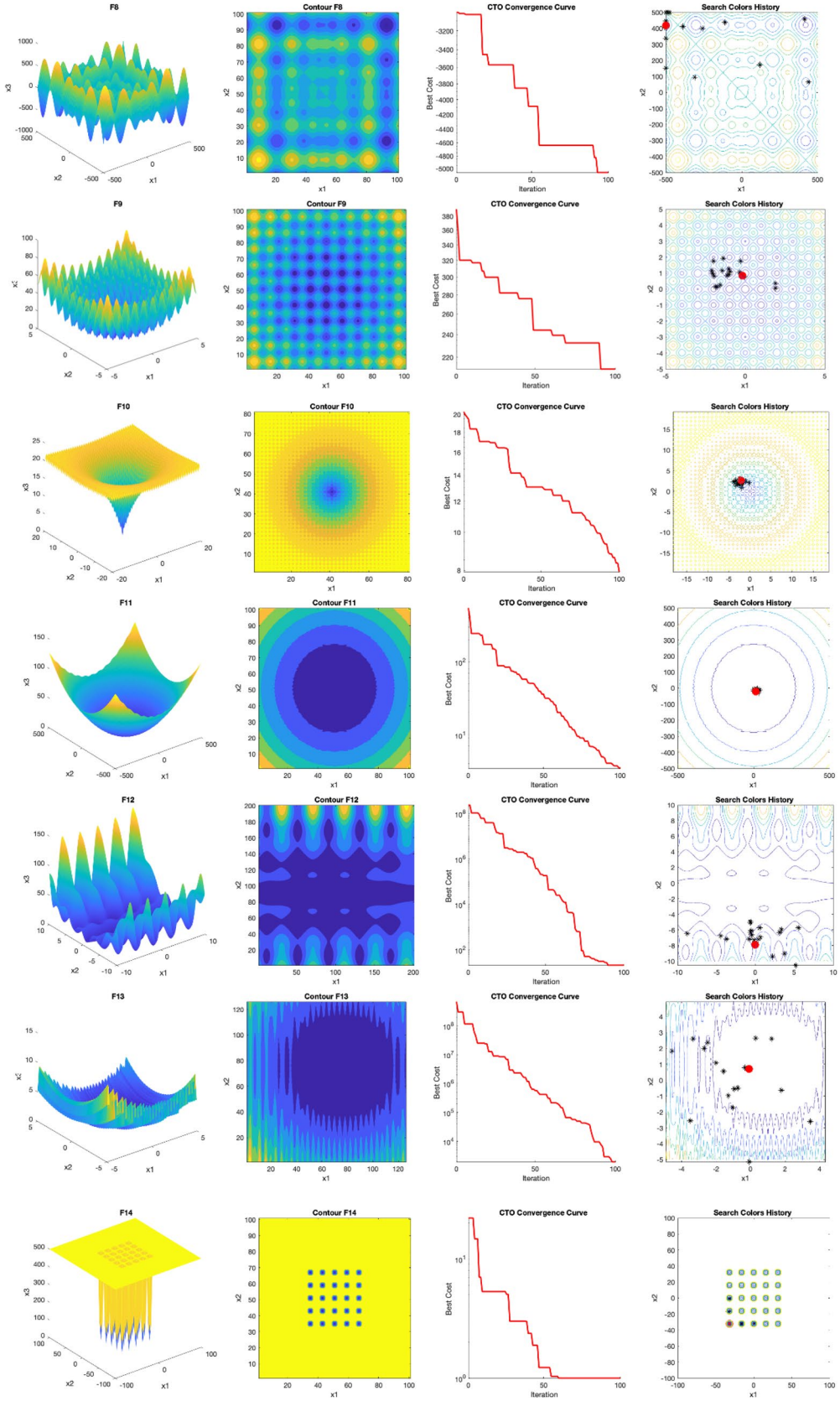

Fig. 10 (continued)

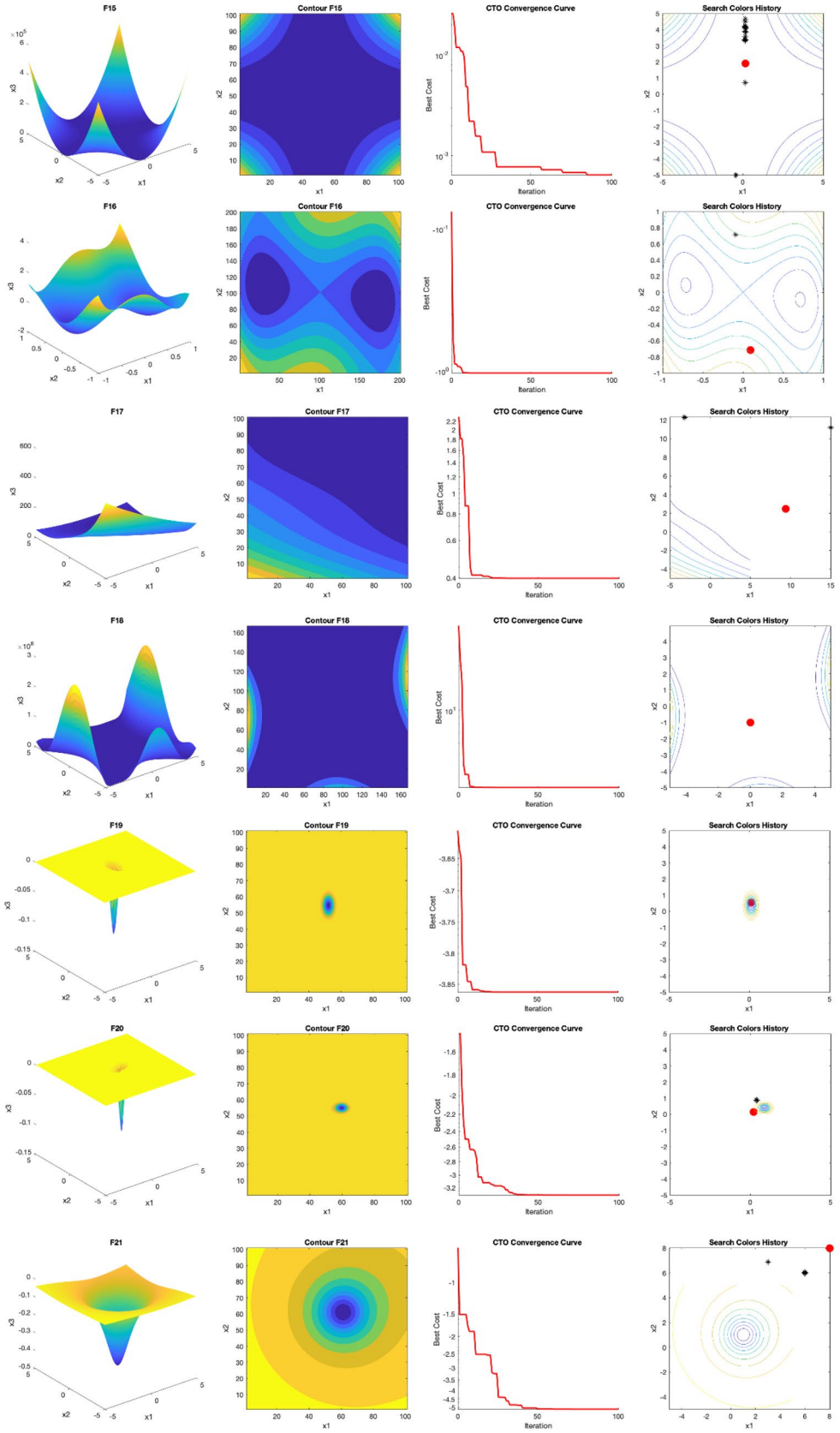

Fig. 10 (continued)

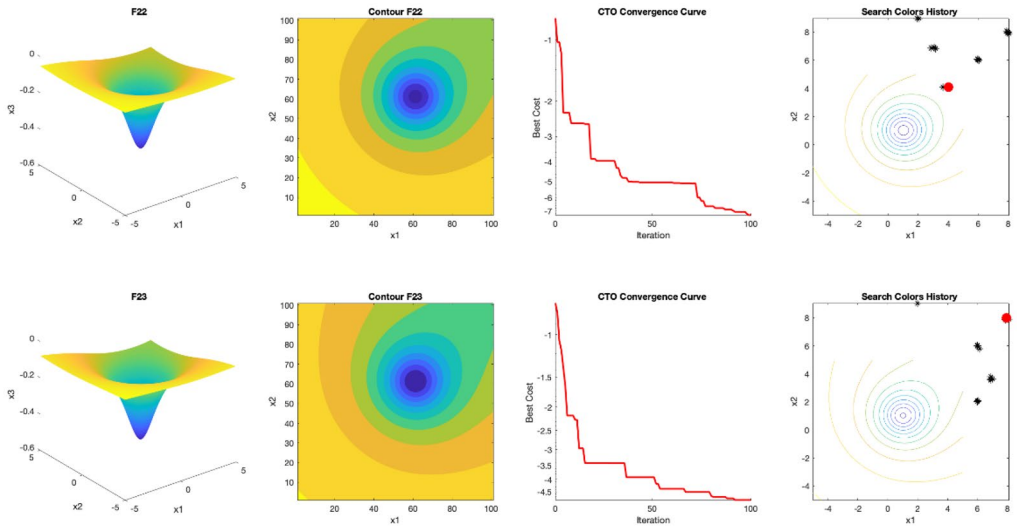

Fig. 10 (continued)

120-bar dome. The dome is subjected to the vertical loading at all free nodes as following: -13.49 kips$(-60$ kN$)$*at* node 1, -6.744 kips$(-30$ kN$)$*at* nodes 2–14, and -2.248 kips$(-10$ kN$)$ at the remaining nodes. Maximum displacements for all nodes in x, y, and z coordinate directions are ± 0.1969in$(\pm 5$mm$)$. The cross-sectional areas are between $0.775\text{in}^2\left(5\text{cm}^2\right)and20\ \text{in}^2(129.032\text{cm}^2)$. This truss is classified as seven symmetric groups. The stress constraints according to the AISC [57] is considered as:

$$\begin{cases} \sigma_i = 0.6F_y\sigma_i \geq 0 \\ \sigma_i = \begin{cases} \left[\left(1-\frac{\lambda_i^2}{2C_c^2}\right)F_y\right]/\left[\frac{5}{3}+\frac{8\lambda_i}{3C_c}-\frac{\lambda_i^3}{8C_c^3}\right] \text{for} \lambda_i > C_c \\ \frac{12\pi^2 E}{23\lambda_i^2} \text{for} \lambda_i \geq C_c \end{cases} \sigma_i < 0 \end{cases} \tag{8}$$

where the modulus of elasticity is described by E, the slenderness ratio is defined by λ_i; C_c is $\sqrt{2\pi^2 E/F_y}$ and the yield stress of steel is F_y.

Table 14 compares the results of the SPO algorithm with the CSS [58], RO [59], CBO [60], CA [61]. The best weight of the present algorithm is 33250.06lb that is the best among the results of other methods. The convergence curve which is obtained for the 120-bar truss problem is shown in Fig. 16. The statistical results in terms of the mean and standard deviation are 33,263.38 lb, and 27.65, respectively. Figure 17 also depicts the existence stress of the best outcomes obtained through SPO.

4.2.3 A 3-bay 15-story frame

This example is a steel frame with 105 members divided into 11 groups as shown in Fig. 18. The modulus of elasticity, the yield stress and unit weight of its constructional steel are 29Msi (200GPa), 36ksi (248.2MPa) and $0.283\text{lb}/\text{in}^3(7.85\text{ton}/\text{m}^3)$, respectively. The effective length factors of the members are calculated as $K_x \geq 0$ for a sway-permitted frame and the out-of-plane effective length factor is specified as $K_y = 1$. For columns, the members are considered non-braced along their length, and for beams, the unbraced length is defined as one-fifth of the span length.

Table 15 presents the best solution for the SPO and some other meta-heuristic algorithms. The SPO has found the lightest design compared to other considered algorithms. The convergence histories of SPO is depicted in Fig. 19. It can be seen that the SPO obtains the best design results which is the lightest among those obtained by the ICA [62], CSS [63], DE [64], WEO [65] and CBO [60]. Results of the standard deviation and average of the SPO, reported in Table 15, are less than those of the other methods. Figures 20 and 21 depict the existing stress ratios and ratio of inter-story drifts to allowable inter-story drifts for the best designs of the present algorithm, respectively.

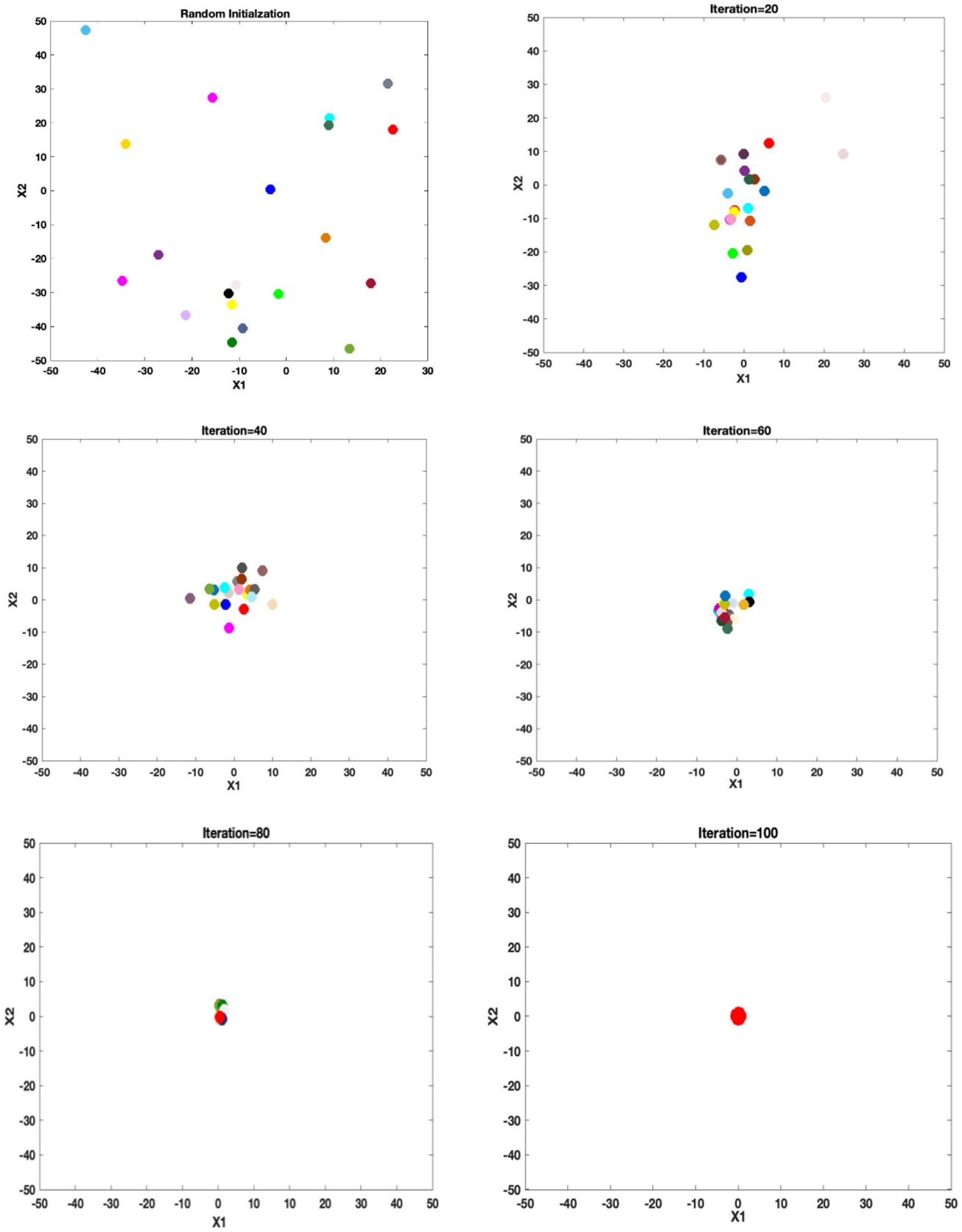

Fig. 11 The position of solutions of the SPO algorithm in different iterations

Table 11 Obtained results from the Wilcoxon test

Compared algorithm	Unimodal functions	Multimodal functions	Fixed-dimension functions
SPO vs. GSA	0.0156	0.3125	0.1562
SPO vs. GWO	0.4688	0.6875	0.9575
SPO vs. SCA	0.0156	0.3125	0.0020
SPO vs. MFO	0.0156	0.0312	0.0312
SPO vs. DA	0.0156	0.0312	0.0469
SPO vs. MVO	0.0312	0.0938	0.0156
SPO vs. WSA	0.3125	0.3750	0.0312

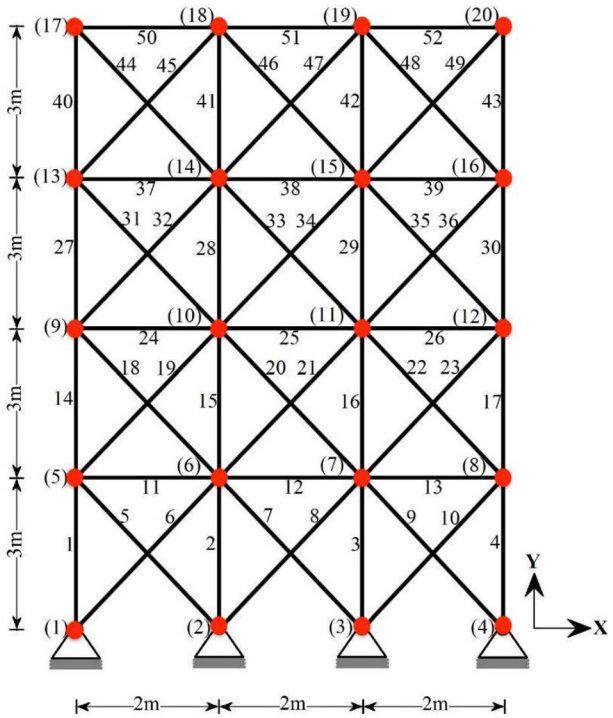

Fig. 12 The 52-bar planar truss structure

Table 12 The available cross-section areas of the AISC code

No	in^2	mm^2	No	in^2	mm^2
1	0.111	71.613	33	3.84	2477.414
2	0.141	90.968	34	3.87	2496.796
3	0.196	126.451	35	3.88	2503.221
4	0.25	161.29	36	4.18	2696.769
5	0.307	198.064	37	4.22	2722.575
6	0.391	252.258	38	4.49	2896.768
7	0.442	285.161	39	4.59	2961.284
8	0.563	363.225	40	4.8	3096.768
9	0.602	388.386	41	4.97	3206.445
10	0.766	494.193	42	5.12	3303.219
11	0.785	506.451	43	5.74	3703.218
12	0.994	641.289	44	7.22	4658.055
13	1	645.1	45	7.97	5141.925
14	1.228	792.256	46	8.53	5503.215
15	1.266	816.773	47	9.3	5999.988
16	1.457	939.998	48	10.85	6999.986
17	1.563	1008.385	49	11.5	7419.43
18	1.62	1045.159	50	13.5	8709.66
19	1.8	1161.288	51	13.9	8967.724
20	1.99	1283.868	52	14.2	9161.272
21	2.13	1374.191	53	15.5	9999.98
22	2.38	1535.481	54	16	10,322.56
23	2.62	1690.319	55	16.9	10,903.204
24	2.63	1696.771	56	18.8	12,129.008
25	2.88	1858.061	57	19.9	12,838.684
26	2.93	1890.319	58	22	14,193.52
27	3.09	1993.544	59	22.9	14,774.164
28	3.13	2019.351	60	24.5	15,806.42
29	3.38	2180.641	61	26.5	17,096.74
30	3.47	2238.705	62	28	18,064.48
31	3.55	2290.318	63	30	19,354.8
32	3.63	2341.931	64	33.5	21,612.86

4.2.4 A 3-bay 24-story frame

This example is a frame structure that consists of 100 joints and 168 members, Fig. 22. The gravity loads and lateral loads are applied to this structure. The material properties of steel are: modulus of elasticity (E) = 29Msi(200GPa), yield stress (Fy) = 36ksi(248.2MPa), and unit weight of the steel (q) = 0.283lb/in^3(7.85ton/m^3). Similar to the previous example, the effective length factors of the members are calculated as $K_x \geq 0$ for a sway-permitted frame and the out-of-plane effective length factor is specified as $K_y = 1$. The columns are considered as non-braced along their length, and the unbraced length for beam members are considered as one-fifth of the span length. To impose fabrication conditions, the beams of the first and third bay except the roof are categorized in one group, which results in four beam groups. The exterior columns are categorized into one group and the interior columns are considered together in another group that changes in every three stories. This grouping results in 16 column groups chosen from 267 W-shape sections and 4 beam groups selected from 37 W14 sections.

Table 13 Comparison of the best design of a 52-bar planar truss

Variables (mm^2)	GA	HS	IACO	MBA	SOS	SPO
1 A1–A4	4658.055	4658.055	4658.055	4658.055	4658.055	4362.809
2 A5–A10	1161.288	1161.288	1161.288	1161.288	1161.288	1183.522
3 A11–A13	645.16	506.451	494.193	494.193	494.193	472.3992
4 A14–A17	3303.219	3303.219	3303.219	3303.219	3303.219	3376.682
5 A18–A23	1045.159	940	939.998	940	940	960.7271
6 A24–A26	494.193	494.193	494.193	494.193	494.193	460.2587
7 A27–A30	2477.414	2290.318	2238.705	2238.705	2238.705	2242.198
8 A31–A36	1045.159	1008.385	1008.385	1008.385	1008.385	985.7185
9 A37–A39	285.161	2290.318	506.451	494.193	494.193	478.5356
10 A40–A43	1696.771	1535.481	1283.868	1283.868	1283.868	1273.351
11 A44-A49	1045.159	1045.159	1161.288	1161.288	1161.288	1151.923
12 A50–A52	641.289	506.451	494.193	494.193	494.193	455.6053
Best Weight (kg)	1970.142	1906.76	1903.183	1902.605	1902.605	**1902.605**
Average weight (kg)	N/A	N/A	N/A	1906.076	N/A	**1903.65**
Standard Deviation	N/A	N/A	N/A	4.09	N/A	**2.975**

Bold values are related to the best results obtained by the selected algorithms

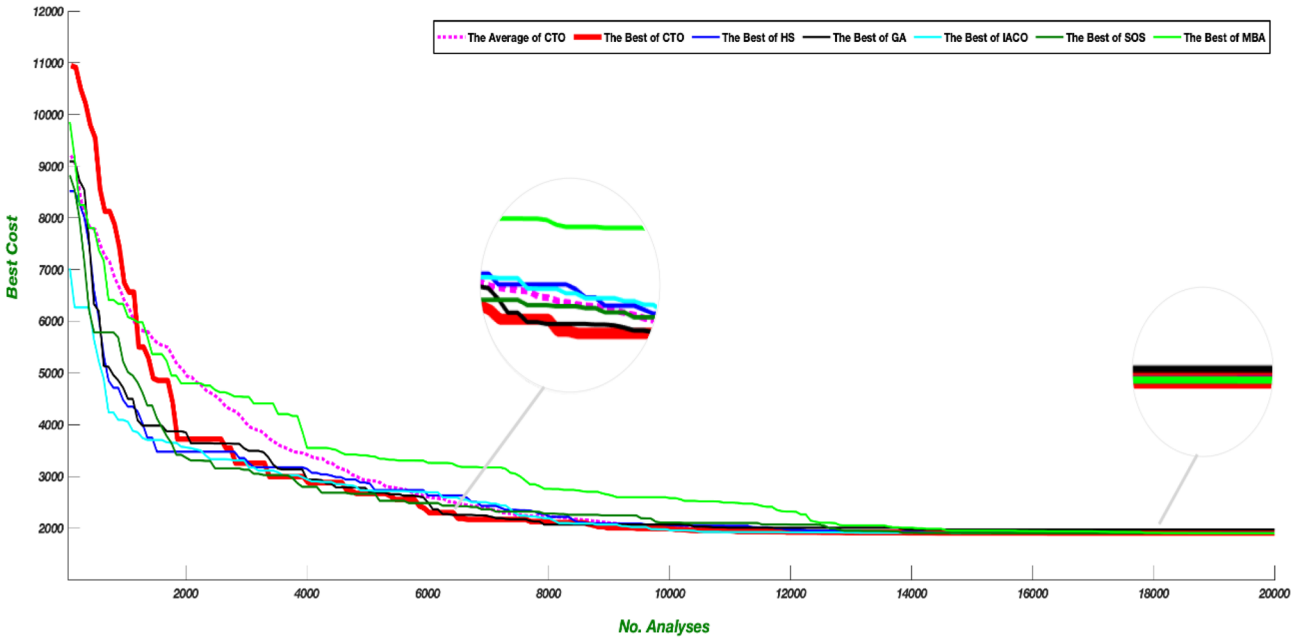

Fig. 13 Convergence history of the SPO and the other utilized algorithms for the 52-bar planar truss structure

As can be seen from Table 16, the SPO has the best performance for this frame structure. Figure 23 shows the convergence histories of SPO. The values of stress corresponding to the optimum design of the SPO is shown in Fig. 24. The best optimal design obtained by the SPO is compared with the results of the ACO [66], HS [67], CBO [14], MFO [68], and DE [64] in Table 16. The results indicate that the optimum design found by the SPO is lighter than the designs obtained by mentioned algorithms. It is observed from Table 16 that the average result of the SPO is less than those of other algorithms. The ratio of inter-story drifts to

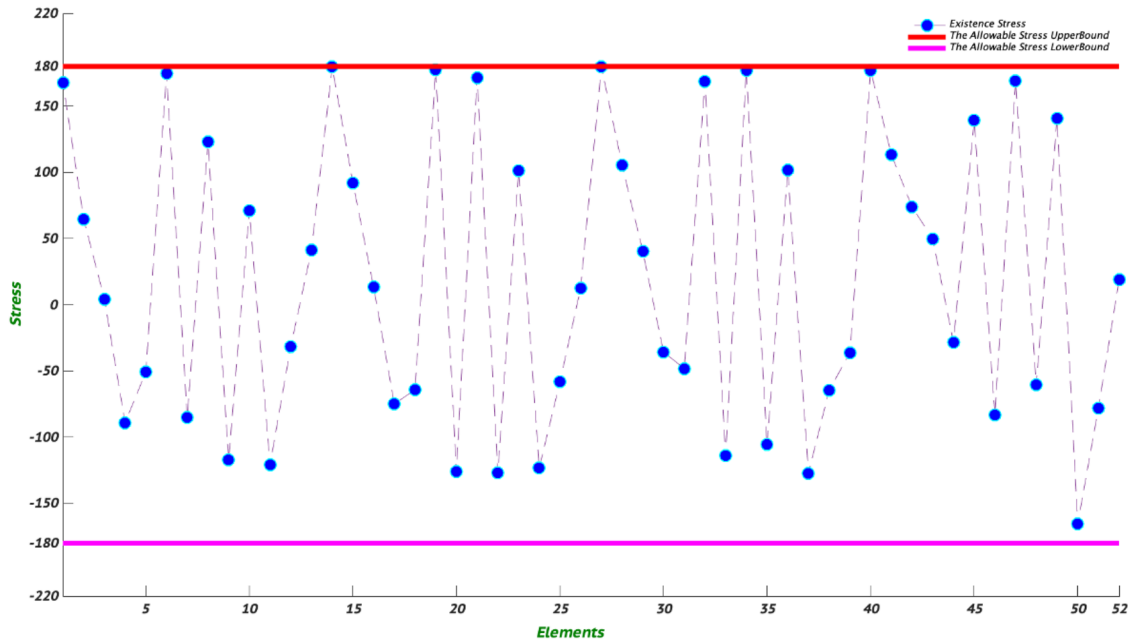

Fig. 14 The limits and existing values of the element stresses for the 52-bar planar truss obtained by the SPO

allowable inter-story drifts for this frame which is obtained by the SPO is illustrated in Fig. 25.

5 Conclusion and future work

This study proposed the SPO algorithm as a new optimization meta-heuristic. the new algorithm is based on the idea of painting a beautiful picture than needs to select the correct color carefully. Three main categories are known for colors and four techniques (analogous, complementary, triadic, and tetradic) were utilized by painters for creating required colors. These categories and techniques are utilized in this paper to generate new colors of paints as solution. for providing a good balancing of exploration and exploitation of the algorithm, these techniques are modified using some stochastic manners. In this way, without increasing the difficulty of the algorithm, a good performance of the algorithm is guaranteed. The resultant algorithm, the SPO, has no parameter to be adjusted and it works with a low initial population size. To sum up, this method is developed based on some important concepts from the field of optimization:

- Its basic fundamental idea has an optimization process in its heart. The painters try to find the best color to paint a beautiful picture and therefore combining different colors should be performed in a heuristic manner. As a result, the formulation of this process will result in an optimization algorithm.
- Two major futures of the optimization algorithms, i.e. exploitation and exploration, are considered and managed in this algorithm. Using the best, average and worst results in the combination process plays the main role in this regard.
- Dividing the colors into three different categories is also useful. When one needs to enhance the exploration, using all these categories together will help and if one wants to improve the exploitation, the best category will be incorporated.
- This algorithm is a population-based method; therefore, it can avoid being trapped in local optima and high exploration is achievable for it compared to the individual-based algorithm.
- Abrupt changes also assist in resolving local optimal stagnations. The position of colors in each iteration is compared to the best colors obtained so far, to provide

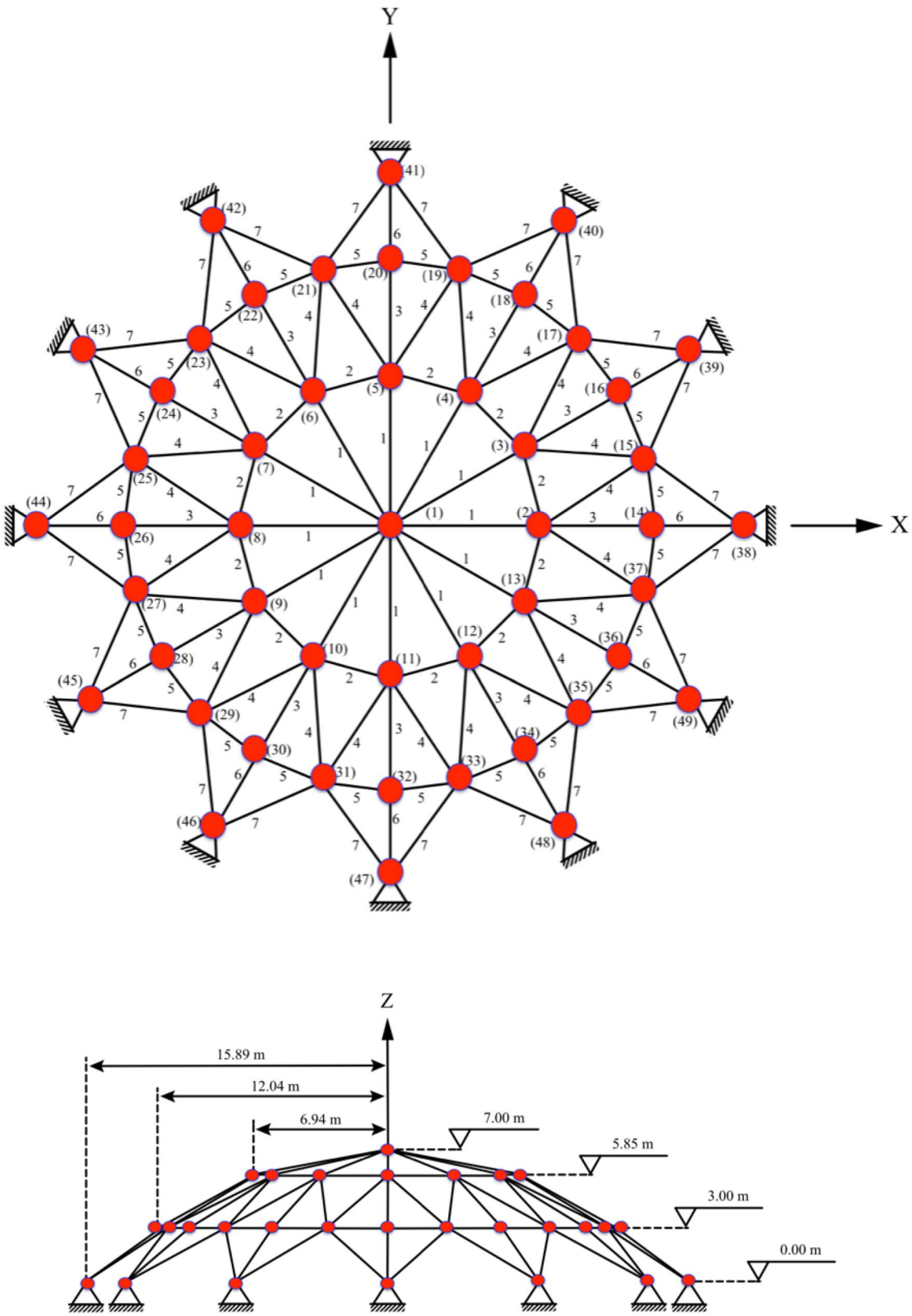

Fig. 15 The 120-bar dome truss structure

Table 14 Comparison of the best design of a 120-bar dome truss

No. Group	CSS	RO	CBO	WEO	CA	SPO
1	3.027	3.030	3.027	3.024	3.025	3.024
2	14.606	14.806	15.172	14.794	14.7652	14.763
3	5.044	5.440	5.234	5.061	5.084	5.100
4	3.139	3.124	3.119	3.135	3.135	3.135
5	8.543	9.021	8.103	8.487	8.438	8.466
6	3.367	3.614	3.416	3.288	3.356	3.292
7	2.497	2.487	2.491	2.496	2.496	2.496
Best Weight (lb)	33,251.9	33,317.8	33,256.15	33,250.24	33,250.95	**33,250.061**
Average weight (lb)	N/A	N/A	33,284.19	33,255.55	N/A	**33,263.38**
Standard Deviation	N/A	354.33	31.40	N/A	N/A	**27.65**

Bold values are related to the best results obtained by the selected algorithms

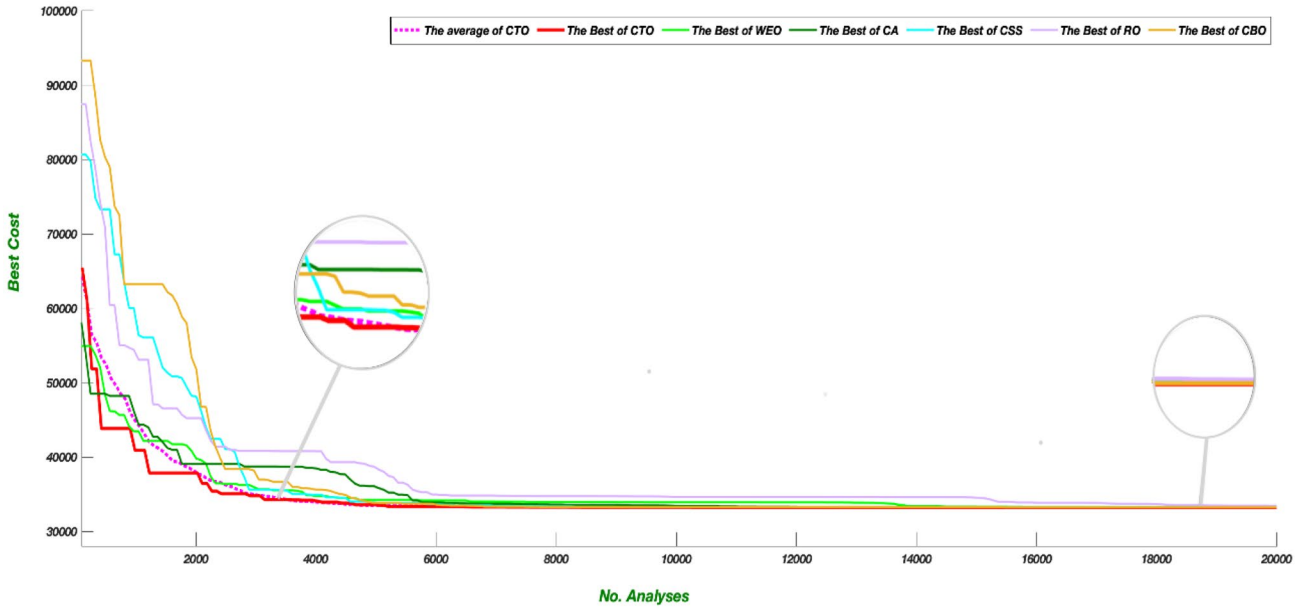

Fig. 16 Convergence history of the SPO and the other utilized algorithms for the 120-bar dome truss structure

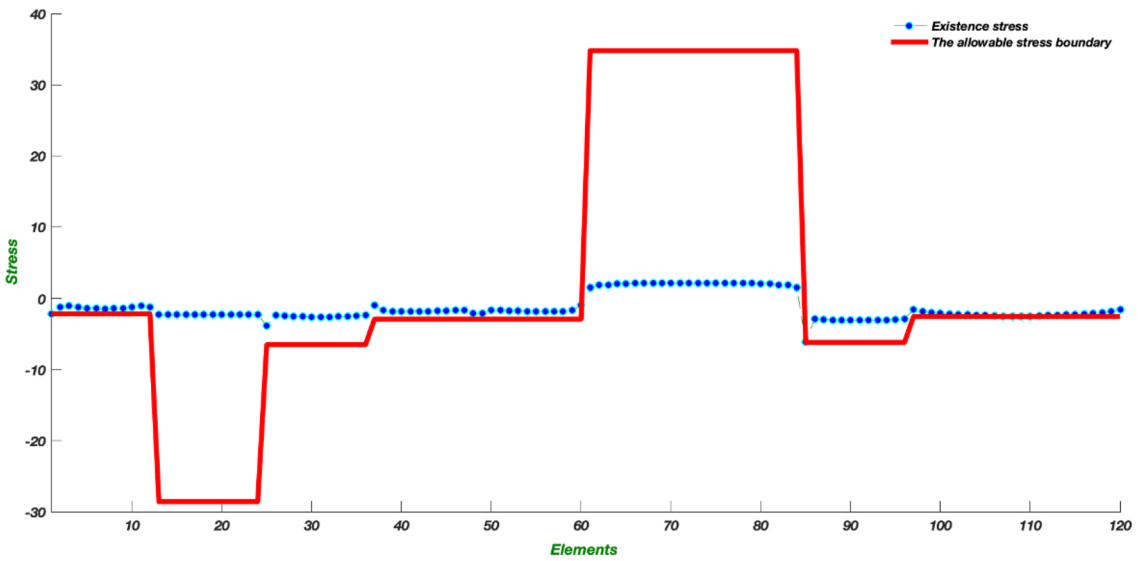

Fig. 17 The limits and existing values of the element stresses for the 120-bar dome truss obtained by the SPO

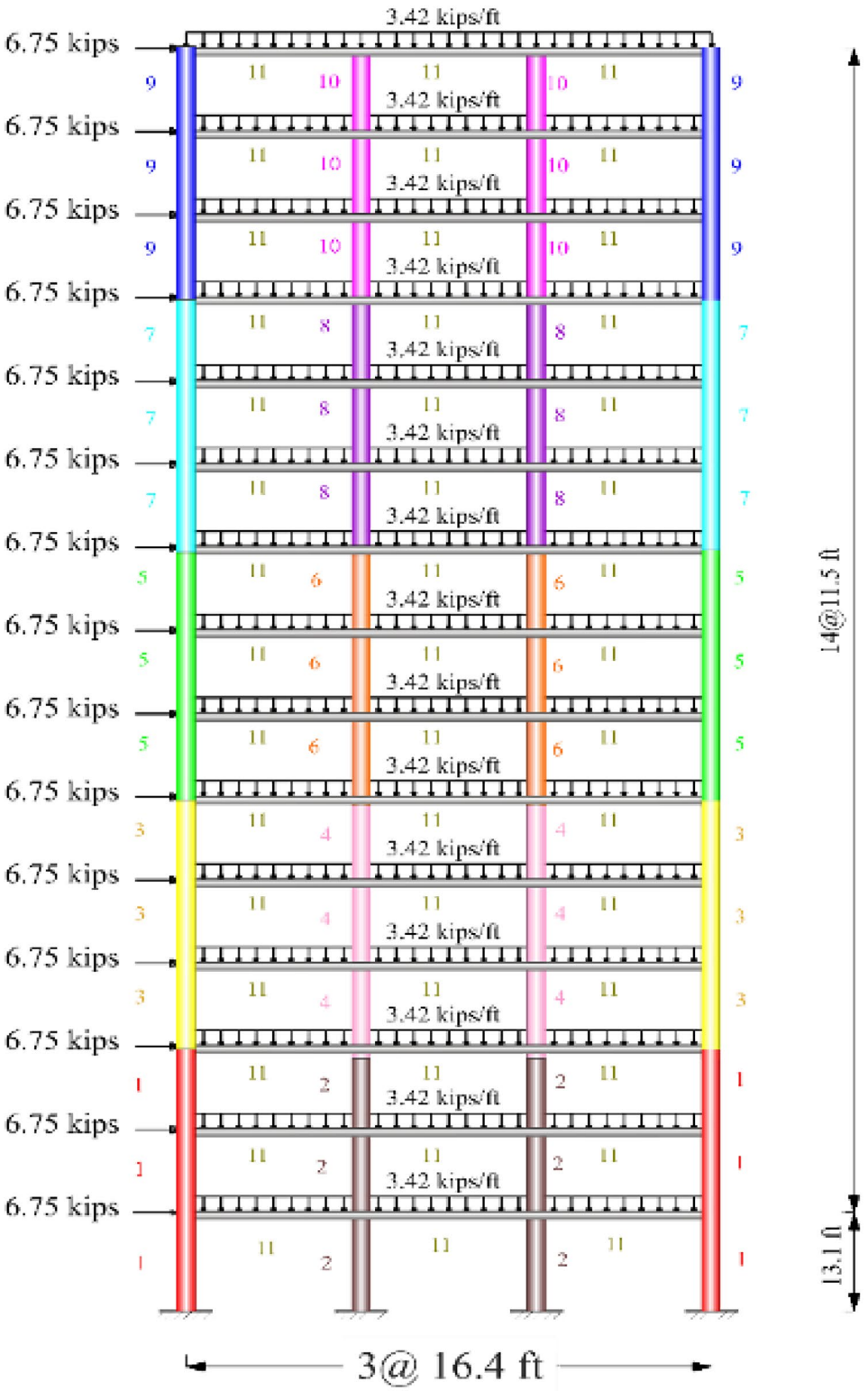

Fig. 18 The 3 bay-15 story frame

Table 15 Comparison of the best design of 3 bay-15 story frame

Group No.	ICA	CSS	DE	WEO	CBO	SPO
1	W24×117	W21×147	W12×87	W14×90	W24×104	W24×104
2	W21×147	W18×143	W36×182	W36×170	W40×167	W36×150
3	W27×84	W12×87	W21×93	W30×90	W27×84	W12×87
4	W27×114	W30×108	W18×106	W24×104	W27×114	W24×104
5	W14×74	W18×76	W18×65	W24×68	W21×68	W27×84
6	W18×86	W24×103	W14×90	W12×87	W30×90	W14×82
7	W12×96	W21×68	W10×45	W18×48	W8×48	W18×60
8	W24×68	W14×61	W12×65	W14×68	W21×68	W14×61
9	W10×39	W18×35	W6×25	W10×33	W14×34	W10×33
10	W12×40	W10×33	W10×45	W16×45	W8×35	W16×36
11	W21×44	W21×44	W21×44	W21×44	W21×50	W21×44
Best Weight (lb)	93,846	92,723	88,878	88,710	93,795	**88,549.7**
Average weight (lb)	N/A	N/A	N/A	90,649	98,738	**90,356**
Standard Deviation	N/A	N/A	N/A	N/A	N/A	**865.36**

Bold values are related to the best results obtained by the selected algorithms

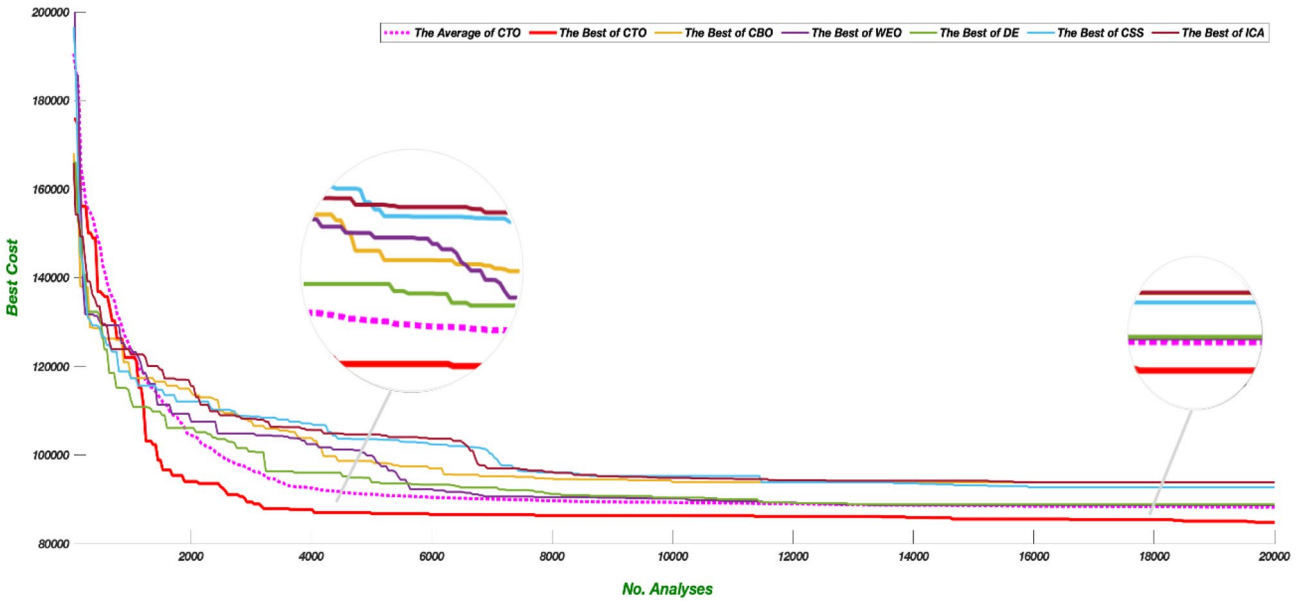

Fig. 19 Convergence history of the SPO and the other utilized algorithms for the 3–15 story frame

a tendency toward the better regions of the search space during optimization.
– The SPO algorithm has no parameter to adjust.

Different mathematical and structural examples are considered for evaluating the performance of the new algorithm. The good results of SPO in benchmark functions (F1–F7) show the performance of the SPO algorithm in terms of exploitation and local optima avoidance. Moreover, SPO has a competitive ability to explore the multimodal topographical spaces with many local optimums and finds the near-global optimum. In addition, the convergence curve of 23 benchmark functions shows SPO's high rate of convergence. The results of the CEC-C06 2019 benchmark

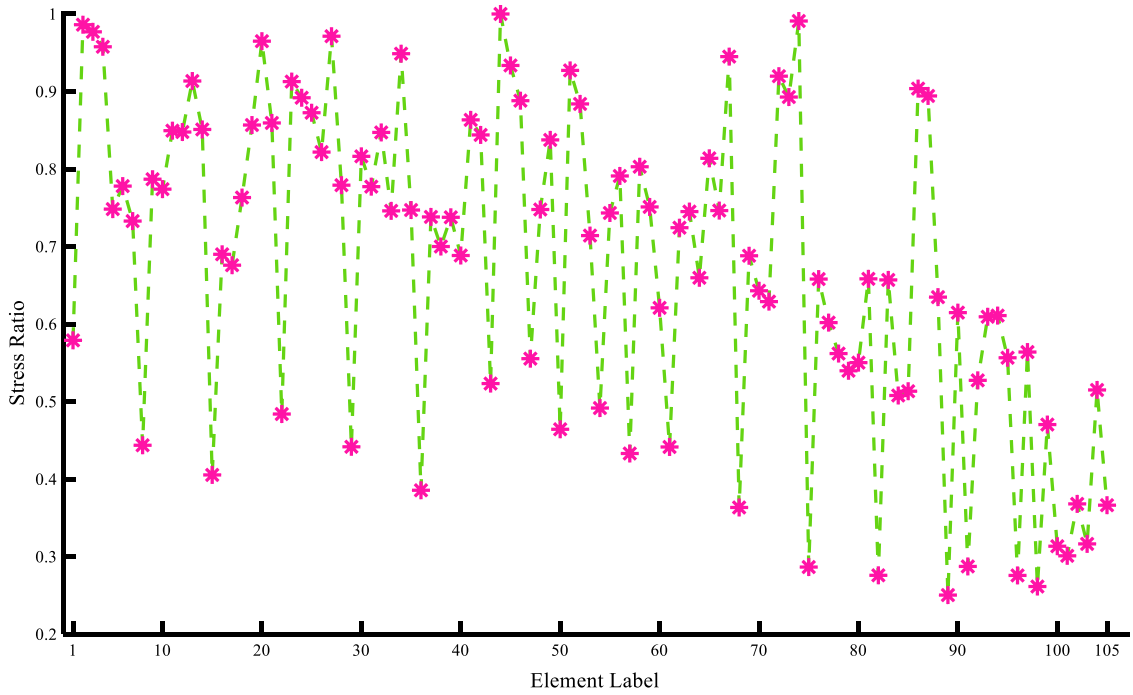

Fig. 20 The limits and existing values of the element stress ratios for the 3 bay-15 story frame obtained by the SPO

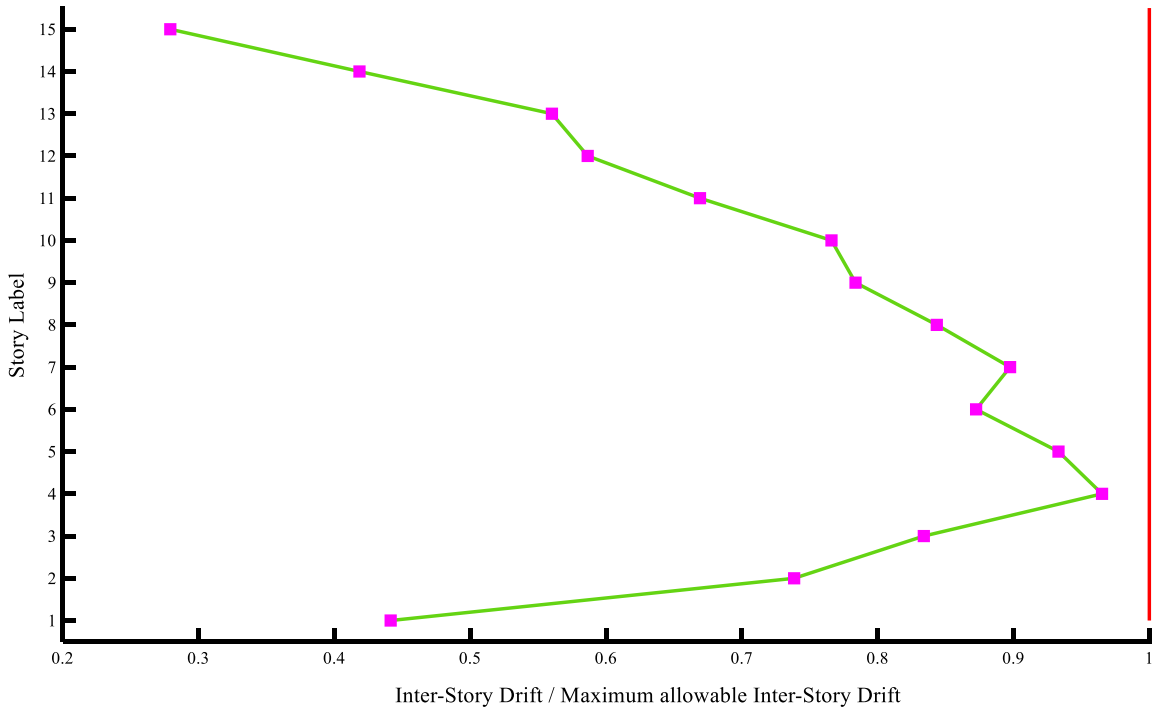

Fig. 21 The inter-story drift to allowable inter-story drift of the 3 bay-15 story frame obtained by the SPO

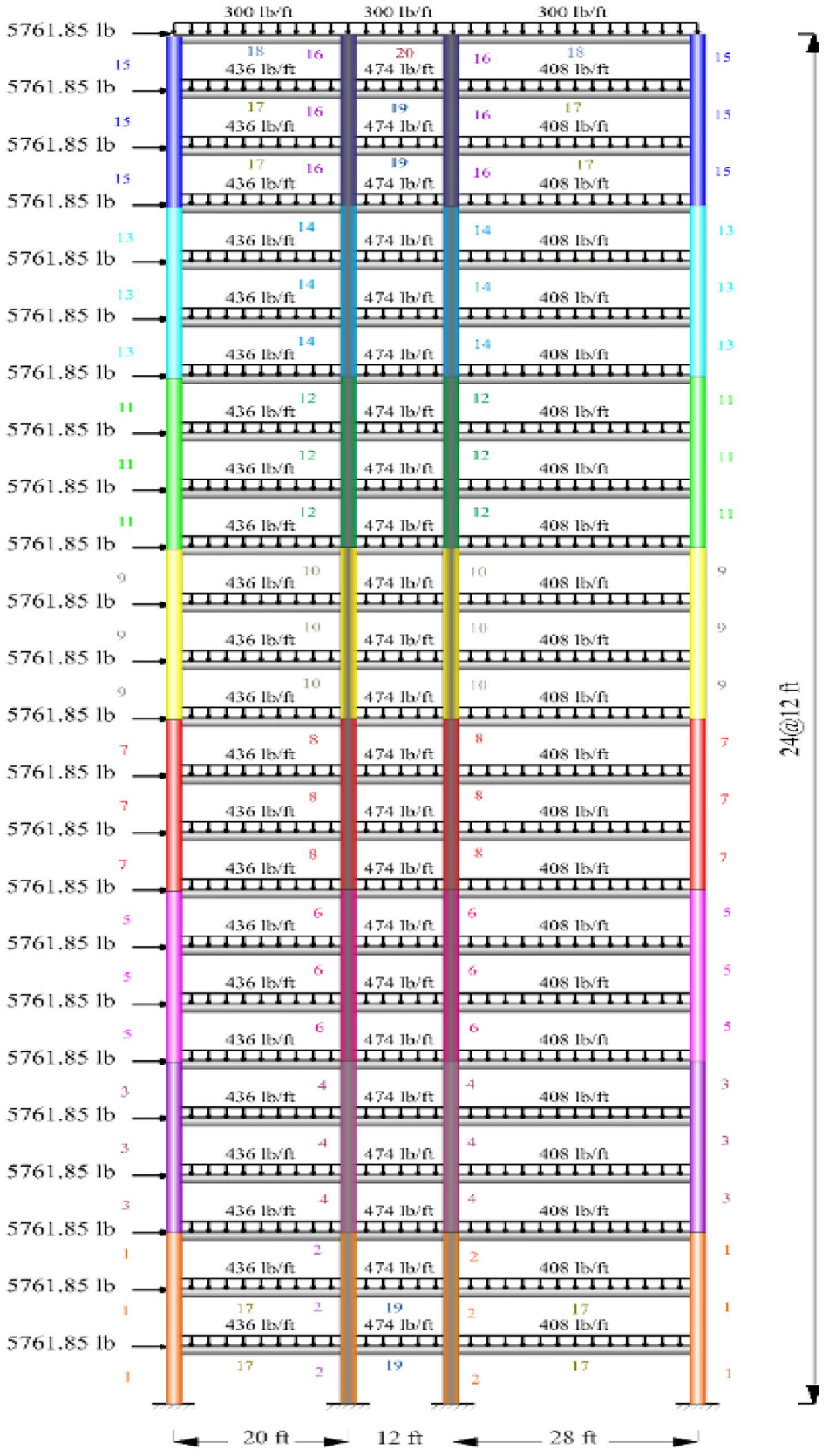

Fig. 22 The 3 bay-24 story frame

Table 16 Comparison of the best design of a 3 bay- 24 story frame

Group No.	ACO	HS	CBO	MFO	DE	**SPO**
1	W30×90	W30×90	W27×102	W30×90	W30×90	W30×90
2	W8×18	W10×22	W8×18	W14×22	W6×20	W21×48
3	W24×55	W18×40	W24×55	W24×55	W21×44	W21×48
4	W8×21	W12×16	W6×8.5	W6×9	W6×9	W14×74
5	W14×145	W14×176	W14×132	W14×159	W14×159	W14×145
6	W14×132	W14×176	W14×120	W14×109	W14×145	W14×120
7	W14×132	W14×132	W14×145	W14×120	W14×132	W14×109
8	W14×132	W14×109	W14×82	W14×74	W14×99	W14×74
9	W14×68	W14×82	W14×61	W14×68	W14×68	W14×53
10	W14×53	W14×74	W14×43	W14×61	W14×61	W14×48
11	W14×43	W14×34	W14×38	W14×38	W14×43	W14×30
12	W14×43	W14×22	W14×22	W14×26	W14×22	W14×30
13	W14×145	W14×145	W14×99	W14×109	W14×109	W14×120
14	W14×145	W14×132	W14×109	W14×109	W14×109	W14×109
15	W14×120	W14×109	W14×82	W14×109	W14×90	W14×99
16	W14×90	W14×82	W14×90	W14×99	W14×82	W14×99
17	W14×90	W14×61	W14×74	W14×82	W14×74	W14×90
18	W14×61	W14×48	W14×61	W14×53	W14×43	W14×61
19	W14×30	W14×30	W14×30	W14×43	W14×30	W14×38
20	W14×26	W14×22	W14×22	W14×26	W14×26	W14×22
Best Weight (lb)	220,465	214,860	215,874	207,793	205,084	**205,055**
Average weight (lb)	229,555	222,620	225,071	N/A	N/A	**215,678**
Standard Deviation	N/A	N/A	N/A	N/A	N/A	**4380.65**

Bold values are related to the best results obtained by the selected algorithms

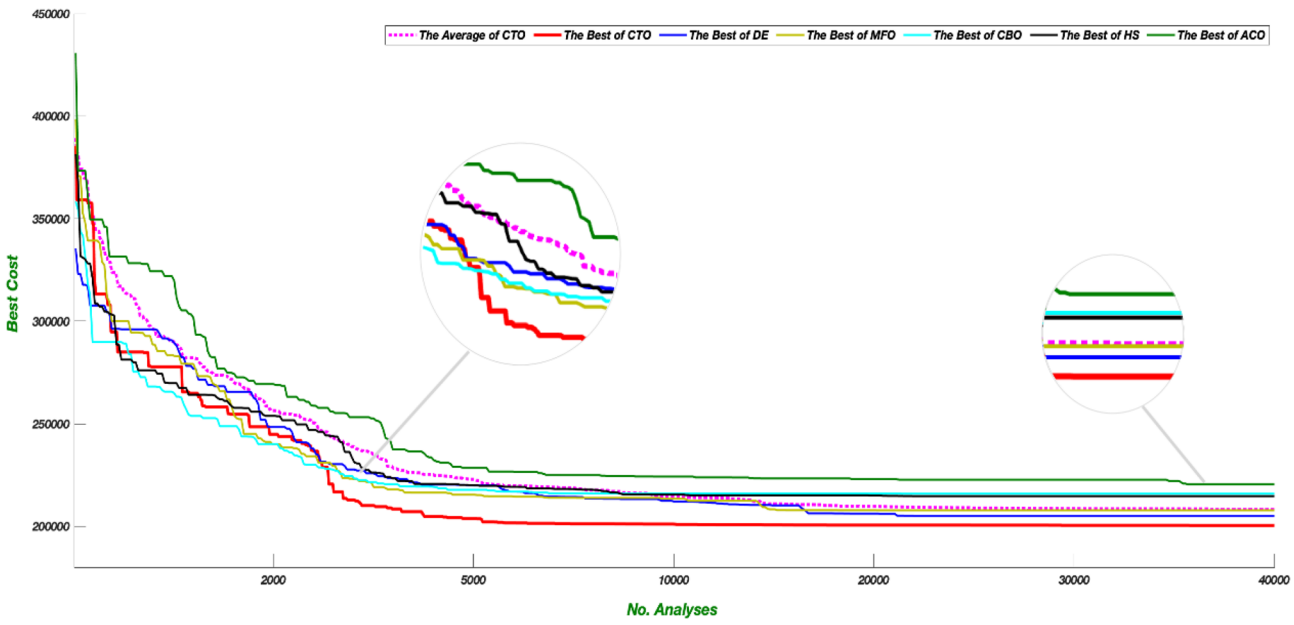

Fig. 23 Convergence history of the SPO and the other utilized algorithms for the 3bay-24 story frame

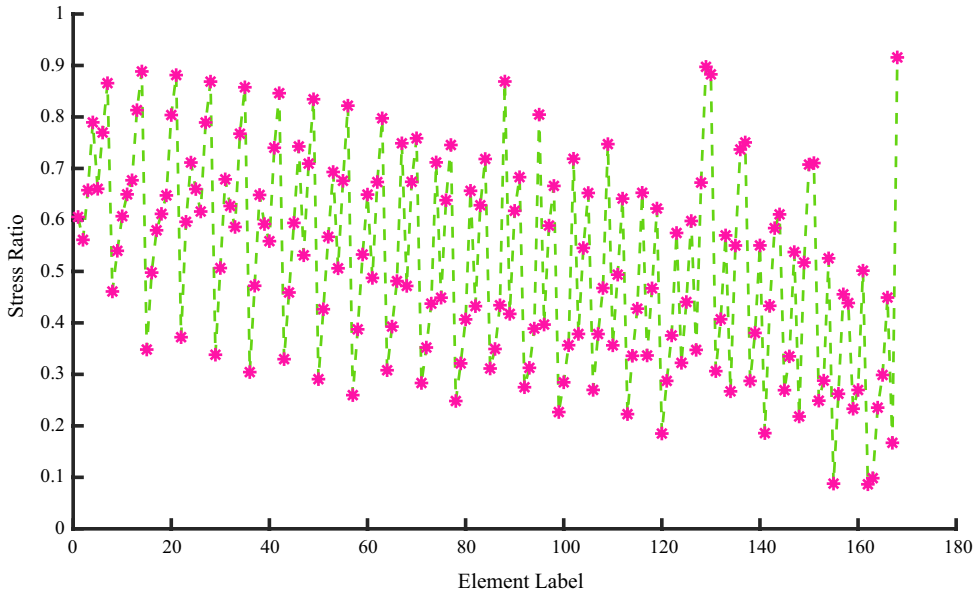

Fig. 24 The limits and existing values of the element stress ratios for the 3 bay-24 story frame obtained by the SPO

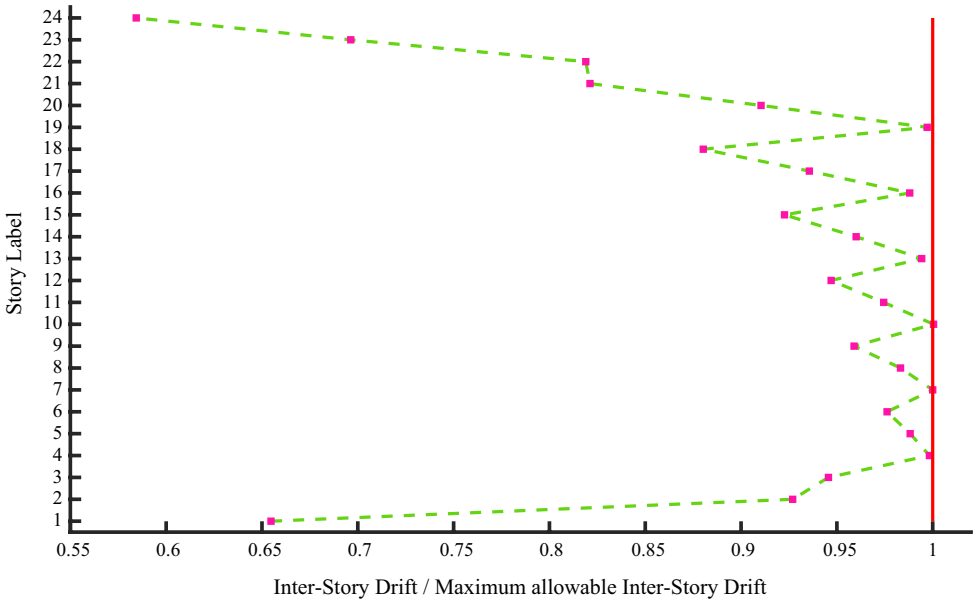

Fig. 25 The inter-story drift to allowable inter-story drift of the 3-bay 24 story frame obtained by the SPO

functions showed the ability of the SPO algorithm in solving the large-scale optimization problems. Finally, the algorithm is applied to four structural designs in engineering and can find the optimal costs for these difficult problems. Future studies will focus on creating different versions of the SPO such as binary, multi-objective versions of this algorithm and using in real large-scale structures problems such as truss and frame with frequency constraints. Besides, various chaotic graphs, hybrid variants can be used to improve the existing algorithm.

References

1. Kaveh A (2017) Advances in metaheuristic algorithms for optimal design of structures, 2nd edn. Springer, Switzerland
2. Talbi E-G (2009) Metaheuristics: from design to implementation, vol 74. Wiley, Hoboken
3. De Jong KA (1975) Analysis of the behavior of a class of genetic adaptive systems, Technical Report
4. Fogel LJ, Owens AJ, Walsh MJ (1966) Artificial intelligence through simulated evolution
5. Holland JH (1992) Adaptation in natural and artificial systems: an introduSPOry analysis with applications to biology, control, and artificial intelligence, MIT press.

6. Goldberg DE (2006) Genetic algorithms, Pearson Education India.
7. Yang X-S et al. (2013) Swarm intelligence and bio-inspired computation: theory and applications, Newnes.
8. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39. https://doi.org/10.1109/MCI.2006.329691
9. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. IEEE. https://doi.org/10.1109/MHS.1995.494215
10. Karaboga D (2010) Artificial bee colony algorithm. Scholarpedia 5(3):6915
11. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
12. Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. Acta Mech 213(3):267–289. https://doi.org/10.1007/s00707-009-0270-4
13. Erol OK, Eksin I (2006) A new optimization method: big bang–big crunch. Adv Eng Softw 37(2):106–111. https://doi.org/10.1016/j.advengsoft.2005.04.005
14. Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 139:18–27. https://doi.org/10.1016/j.compstruc.2014.04.005
15. Rashedi E, Nezamabadi-Pour H, Saryazdi SJIS (2009) GSA: a gravitational search algorithm 179(13): 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
16. Van Laarhoven PJ, Aarts EH (1987) Simulated annealing. Simulated annealing: theory and applications. Springer, Berlin, pp 7–15
17. Li S et al (2020) Slime mould algorithm: a new method for stochastic optimization. Future Generation Comput Syst. https://doi.org/10.1016/j.future.2020.03.055
18. Kaveh A, Dadras Eslamlou A (2020.) Water strider algorithm: A new metaheuristic and applications. Structures, Elsevier, https://doi.org/10.1016/j.istruc.2020.03.033
19. Kahraman HT, Aras S, Gedikli E (2020) Fitness-distance balance (FDB): a new selection method for meta-heuristic search algorithms. Knowl-Based Syst 190:105169. https://doi.org/10.1016/j.knosys.2019.105169
20. Kaveh A, Talatahari S, Khodadadi N (2019) Hybrid invasive weed optimization-shuffled frog-leaping algorithm for optimal design of truss structures. Iran J Sci Technol Trans Civ Eng 2019:1–16. https://doi.org/10.3311/PPci.14576
21. Pijarski P, Kacejko P (2019) A new metaheuristic optimization method: the algorithm of the innovative gunner (AIG). Eng Opt 51(12):2049–2068. https://doi.org/10.1080/0305215X.2019.1565282
22. Fathollahi-Fard AM, Hajiaghaei-Keshteli M, Tavakkoli-Moghaddam (2020) Red deer algorithm (RDA): a new nature-inspired meta-heuristic. Soft Computing, 2020: 1–29. https://doi.org/10.1007/s00500-020-04812-z
23. Mafarja M et al. Dragonfly algorithm: theory, literature review, and application in feature selection, in Nature-Inspired Optimizers. 2020, Springer, Berlin, pp 47–67. https://doi.org/10.1007/978-3-030-12127-3_4
24. Kaveh A, Dadras Eslamlou A (2020) Metaheuristic optimization algorithms in civil engineering: new applications. Springer, Berlin
25. Kaveh A, Ilchi Ghazaan M (2018) Meta-heuristic algorithms for optimal design of real-size structures. Springer, Berlin
26. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68. https://doi.org/10.1177%2F003754970107600201
27. Zaeimi M, Ghoddosian A (2020) Color harmony algorithm: an art-inspired metaheuristic for mathematical function optimization. Soft Comput 2020: 1–40. https://doi.org/10.1007/s00500-019-04646-4
28. Matsuda YJAS (1995) Color design 2(4):10
29. Tokumaru M, Muranaka N, Imanishi S (2002) Color design support system considering color harmony. In: 2002 IEEE world congress on computational intelligence. 2002 IEEE international conference on fuzzy systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291). 2002. IEEE. https://doi.org/10.1109/FUZZ.2002.1005020
30. Cheng S, Shi Y (2011) Diversity control in particle swarm optimization. In: 2011 IEEE symposium on swarm intelligence. IEEE. https://doi.org/10.1109/SIS.2011.5952581
31. Wolpert DH, W.G.J.I.t.o.e.c. (1997) Macready, No free lunch theorems for optimization. 1(1): p. 67–82. https://doi.org/10.1109/4235.585893
32. Wool LE et al (2015) Salience of unique hues and implications for color theory. J Vis 15(2):10–10. https://doi.org/10.1167/15.2.10
33. Parkhurst C, Feller RL (1982) Who invented the color wheel? Color Res Appl 7(3):217–230. https://doi.org/10.1002/col.5080070302
34. Feisner EA (2006) Colour: how to use colour in art and design. Laurence King Publishing.
35. Westland S et al (2007) Colour harmony. Colour: Design Creativity 1(1):1–15
36. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3(2):82–102. https://doi.org/10.1109/4235.771163
37. Digalakis JG, Margaritis KG (2001) On benchmarking functions for genetic algorithms. Int J Comput Math 77(4):481–506
38. Molga M, Smutnicki C (2005) Test functions for optimization needs. Test functions for optimization needs, 101.
39. Yang X-S Test problems in optimization. arXiv preprint arXiv:1008.0549, 2010.
40. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
41. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133. https://doi.org/10.1016/j.knosys.2015.12.022
42. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249. https://doi.org/10.1016/j.knosys.2015.07.006
43. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513. https://doi.org/10.1007/s00521-015-1870-7
44. Price K et al (2018) The 100-digit challenge: Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization. Nanyang Technological University, Singapore
45. Rahman CM, Rashid TA (2019) Dragonfly algorithm and its applications in applied science survey. Comput Intell Neurosci. https://doi.org/10.1155/2019/9293617
46. Mohammed HM, Umar SU, Rashid TA (2019) A systematic and meta-analysis survey of whale optimization algorithm. Comput Intell Neurosci. https://doi.org/10.1155/2019/8718571
47. Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. Soft Comput 23(3):715–734. https://doi.org/10.1007/s00500-018-3102-4
48. Ahmed AM, Rashid TA, Saeed SAM (2020) Cat swarm optimization algorithm: a survey and performance evaluation. Comput Intell Neurosci. https://doi.org/10.1155/2020/4854895
49. Mirjalili S et al (2017) Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems 114:163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002

50. Van den Bergh F, Engelbrecht AP (2006) A study of particle swarm optimization particle trajeSPOries. Inf Sci 176(8):937–971. https://doi.org/10.1016/j.ins.2005.02.003
51. Wu S-J, Chow P-T (1995) Steady-state genetic algorithms for discrete optimization of trusses. Comput Struct 56(6):979–991. https://doi.org/10.1016/0045-7949(94)00551-D
52. Lee KS et al (2005) The harmony search heuristic algorithm for discrete structural optimization. Eng Opt 37(7):663–684. https://doi.org/10.1080/03052150500211895
53. Kalatjari VR, Talebpour MH (2017) An improved ant colony algorithm for the optimization of skeletal structures by the proposed sampling search space method. Periodica Polytechnica Civ Eng 61(2):232–243. https://doi.org/10.3311/PPci.9153
54. Sadollah A et al (2012) Mine blast algorithm for optimization of truss structures with discrete variables. Comput Struct 102:49–63. https://doi.org/10.1016/j.compstruc.2012.03.013
55. Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112. https://doi.org/10.1016/j.compstruc.2014.03.007
56. Soh CK, Yang J (1996) Fuzzy controlled genetic algorithm search for shape optimization. J Comput Civ Eng 10(2):143–150. https://doi.org/10.1061/(ASCE)0887-3801(1996)10:2(143)
57. American Institute of Steel Construction (AISC) Manual of steel construction: allowable stress design. 1989.
58. Kaveh A, Talatahari S (2010) Optimal design of skeletal structures via the charged system search algorithm 41(6):893–911. https://doi.org/10.1007/s00158-009-0462-5
59. Kaveh A, Khayatazad M (2012) A new meta-heuristic method: ray optimization. Comput Struct 112:283–294
60. Kaveh A, Mahdavi VR (2015) Colliding bodies optimization: extensions and applications. Springer, Berlin
61. Jalili S, Hosseinzadeh Y (2015) A cultural algorithm for optimal design of truss structures. Latin Am J Solids Struct 12(9):1721–1747. https://doi.org/10.1590/1679-78251547
62. Kaveh A, Talatahari S (2010) Optimum design of skeletal structures using imperialist competitive algorithm. 88(21–22): 1220–1229. https://doi.org/10.1016/j.compstruc.2010.06.011
63. Kaveh A, Talatahari S (2012) Charged system search for optimal design of frame structures. 12(1): 382-393. https://doi.org/10.1016/j.asoc.2011.08.034
64. Kaveh A, Farhoudi N (2013) A new optimization method. Dolphin Echolocation 59:53–70. https://doi.org/10.1016/j.advengsoft.2013.03.004
65. Kaveh A, Bakhshpoori T (2016) An accelerated water evaporation optimization formulation for discrete optimization of skeletal structures. Comput Struct 177:218–228. https://doi.org/10.1016/j.compstruc.2016.08.006
66. Camp CV, Bichon BJ, Stovall SP (2005) Design of steel frames using ant colony optimization. J Struct Eng 131(3):369–379. https://doi.org/10.1061/(ASCE)0733-9445(2005)131:3(369)
67. Degertekin SO (2008) Optimum design of steel frames using harmony search algorithm. Struct Multidisciplinary Opt 36(4):393–401. https://doi.org/10.1007/s00158-007-0177-4
68. Gholizadeh S, Davoudi H, Fattahi F (2017) Design of steel frames by an enhanced moth-flame optimization algorithm. Steel Compos Struct 24(1):129–140. https://doi.org/10.12989/scs.2017.24.1.129